MASTER'S THESIS

Improving student succes prediction using (Bayesian) Machine learning

Groenveld, L. (Lauran)

Award date:
2020

Improving Student Success Prediction using (Bayesian) Machine Learning

MSc Thesis

L. Groenveld, student no.

April 21, 2020

[This page is intentionally left blank]

Student	Lauran Groenveld
Student number	
Thesis title	Improving Student Success Prediction using (Bayesian) Machine Learning
Graduation committee:	Dr. A. J. Hommersom, Open University of the Netherlands
	Dr. T. van Laarhoven, Open University of the Netherlands
Supervisor	Dr. A. J. Hommersom, Open University of the Netherlands
Degree programme	Open University of the Netherlands, Faculty of Management, Science and Technology Master's Programme in Software Engineering
Course code	IM9906

Contents

1 Summary

The prediction of student success probabilities early on in the schoolyear is valuable for different departments in institutions for Secondary Education in the Netherlands. Based on historical data, using the course marks of the students, the success chances may be evaluated with a supervised learning algorithm for classification. However, the straightforward use of such algorithms for group-wise predictive models introduces an issue which we will term the *Group Performance Problem*.

This problem states that if the probability distribution of the features in the new group significantly differs from that distribution in the training groups, and it is known that this difference will lead to different interventions with respect to the individuals in the new group, a traditional predictive model may fail to provide a reliable prediction. The strength of this effect may be quantified in the *Group Performance Factor* (GPF).

Multiple strategies for dealing with this problem are proposed and compared: incorporating additional or transformed features in traditional models, and creating Bayesian models using Monte Carlo Markov Chain (MCMC) sampling. In the *Group Difference model*, continuous features relative to the own group mean are used. In the *Bayesian Group model*, in addition to the student course marks, the historical group success ratios are incorporated as evidence, while the *Bayesian Lambda model* determines a multiplying factor for increasing individual success probabilities in the case of a strong group effect.

These models are tested using synthetic data simulations with varying strengths of the GPF. Furthermore, a case study with data of a secondary school in the Netherlands is used to test the models. As individual success probabilities are more important than the predicted class labels, the strictly proper *Brier score* is used to determine the performance of all models.

We were able to successfully improve the prediction using the different strategies in the synthetic dataset simulations. In the case of very strong group effects, the Group Difference model or the Bayesian Lambda model turned out to be the most successful strategy. When no significant group effect is expected, standard models like Logistic Regression are the best choice. In all other cases, including the scenario when estimating the group effect by domain experts is not available, the Bayesian Group model may provide robust results. The case study using data from our example school appeared to support these conclusions, but the results were not statistically significant.

Future work may be focused on the definition of the GPF, possibly examining the *group effect* (covariate shift) and the *intervention effect* separately. Furthermore, the strategies and corresponding models should be tested in other scenarios, and additional strategies may be formulated and examined. Finally, future work with respect to the runtime of predictions is recommended as this may be important in other domains, e.g. online learning.

2 Introduction

Educational Institutions generally have a need for accurate predictions of the student success probabilities. In Secondary Education in the Netherlands, upper and middle management are interested in these predictions for multiple reasons.

In this type of education, provided at Secondary Schools, pupils are placed at a certain grade and educational level based on age, intelligence and previous achievements. At the end of every schoolyear, a decision is made for the placement of students in the next schoolyear. With satisfactory marks for the courses followed, the student will pass, and generally continue in the next grade at the same level after the summer. If marks do not meet the *passing standard* set by the school, the student may repeat the class, continue at the next grade at a lower level, leave the school or be placed in the next grade despite not meeting this standard. The decision is reserved for members of the *passing meeting*, usually consisting of the teachers and the department leadership.

2.1 Goals of Student Success Prediction

The prediction of success probabilities, the chance for a pupil to pass at the end of the schoolyear, is valuable for different departments. There are roughly two concerns.

- **Individual Success Prediction**
  Middle management and counsellors are concerned with the success chances of the students belonging to their department. These predictions may give rise to certain interventions, aimed at improving the marks (and therefore the chance of passing) of a pupil. Examples of these interventions are parent meetings, composing a personal plan of action, tutoring, provide training for performance anxiety, and other measures aimed at improving the social-emotional or pedagogic circumstances.
  As the schoolyear is divided in *terms* (usually 2-4, typically 3), the students' results at the end of a term trigger these interventions. However, because resources are limited, interventions are obviously best targeted at students with high chances of failing the schoolyear. Later on, resources may better be used for students that turn out to be on the edge of failing/passing (like a success chance of 50 percent) for the best return on investment with respect to these interventions. Individual success predictions may therefore be used for monitoring and decision making processes at these departments.

- **Success Ratio Prediction**
  Upper management and the planning department are concerned with the predicted success ratios for each student department (combination of level and grade). Early on, the staff occupation for the next schoolyear should be established. The composition of the teaching team is highly dependent on the expected student department size in the next schoolyear. As the student success ratios determine future student department sizes, they are of great value for the management and planning teams. In this case, the cumulative probability of all students being placed in a certain department

is more important than the actual expected placement of an individual pupil.

The first goal, predicting the success chances of individual students, requires a high accuracy. The actual chance of passing becomes particularly important after the first term, as available resources are not necessarily best targeted at the set of students with the highest risk of failing, but may instead be used for students that are at the edge of failing/passing. Because these students only need a small boost to pass at the end of the schoolyear, the return on investment with respect to interventions is high. Therefore, an adequate estimate of the individual chances of each student is desirable.
Accuracy is also highly important for the second goal: success ratio prediction. A proper estimate of the expected success ratio for each department is of great value for the organizational and financial planning of the next (and subsequent) schoolyears. Early on in the schoolyear, there is typically need for a forecast with respect to the number of students for each department, both for the next year and a multi-annual prediction.

2.2 Management Tools

Schools may use tools to monitor student progress and provide success predictions. These software solutions, using data from the *School Administration System (SAS)*, are part of a group of tools called *Learning Analytics* [Claas L., 2017]. The success predictions of students of these tools are traditionally performed in one of two ways:

- **Manual evaluation**
  In this case, employees of the schools (e.g. counsellors, team leaders), manually evaluate the chances of each pupil. By entering their expectation, the management tool is able to demonstrate aggregated data to the management team. Obviously, this method is highly dependent on the skill, experience and engagement of the employees concerned with the task of evaluating the success chances of the students. In practice, the quality of these expectations is not satisfactory and varies considerably.

- **Evaluation of meeting the passing standard**
  Another option is to check the course marks of the student, and evaluate whether or not the passing standard is met. The passing standard typically prescribes the minimum average and maximum insufficient course marks needed by the end of the schoolyear to pass. Because this standard is established in advance and is published by the school, the evaluation is easily automated. However, there is an important caveat when using evaluation of the passing standard as a prediction for student success, which will be described in the next subsection.

Results of these success predictions may be presented in a dashboard of the application. Stakeholders, like management and counsellors, are able to use the data for their monitoring and planning goals..

2.3 Using the passing standard

Student grades obviously fluctuate during the schoolyear. We will demonstrate that estimating student success chances by evaluation of the grades with the passing standard early on in the schoolyear will significantly underestimate their chances.
To illustrate this last observation, we will introduce some data of an existing secondary school in the Netherlands. The passing standard is well-defined at this institution, though may be different for each department (combination of level and grade). The schoolyear is divided in three terms, each one consisting of about 12 school weeks. Management tools should sufficiently predict student success ratio even after the first term, as the organizational planning of the next schoolyear is already started by that time. Furthermore, individual success prediction after this first term should be accurate as most interventions are started in the second term.

Evaluating the passing standard for each student after the first term, as is traditionally used for student success ratio prediction at this school, significantly underestimates student success chances (table 2.1).

Transition year	Term 1	Term 2	Term 3	Actual
2015	0.74	0.79	0.81	0.89
2016	0.75	0.82	0.86	0.89
2017	0.69	0.79	0.84	0.87
2018	0.67	0.78	0.85	0.89

Table 2.1: Success ratio after each term based on evaluation of the passing standard, and actual end-of-year success ratio (all departments).

Two conclusions may be derived from the data in this table. First, the ratio of students meeting the passing standard increases over the schoolyear. Fortunately, results tend to improve (which may partly be due to student interventions). However, the other conclusion with respect to using the passing standards for student success prediction at this institution is more important:

The actual ratio of students passing is considerably higher than the ratio of students meeting the passing standard

This conclusion is valid even in the last term (end of the schoolyear). Apparently, part of the students are placed in the next grade despite not meeting the passing standard. Based on this observation, we may conclude that using the passing standard directly to predict student success prediction is not accurate enough to meet the goals described in section 2.1.

2.4 The potential of Machine Learning

Using evaluation of the passing standard for student success prediction underestimates their chances. However, there is an obvious correlation between the grades of students and the chance of passing at the end of the schoolyear. For our example school, the scatterplot of figure 2.1 demonstrates the correlation

between grades of the first and the last term, and the actual student success.

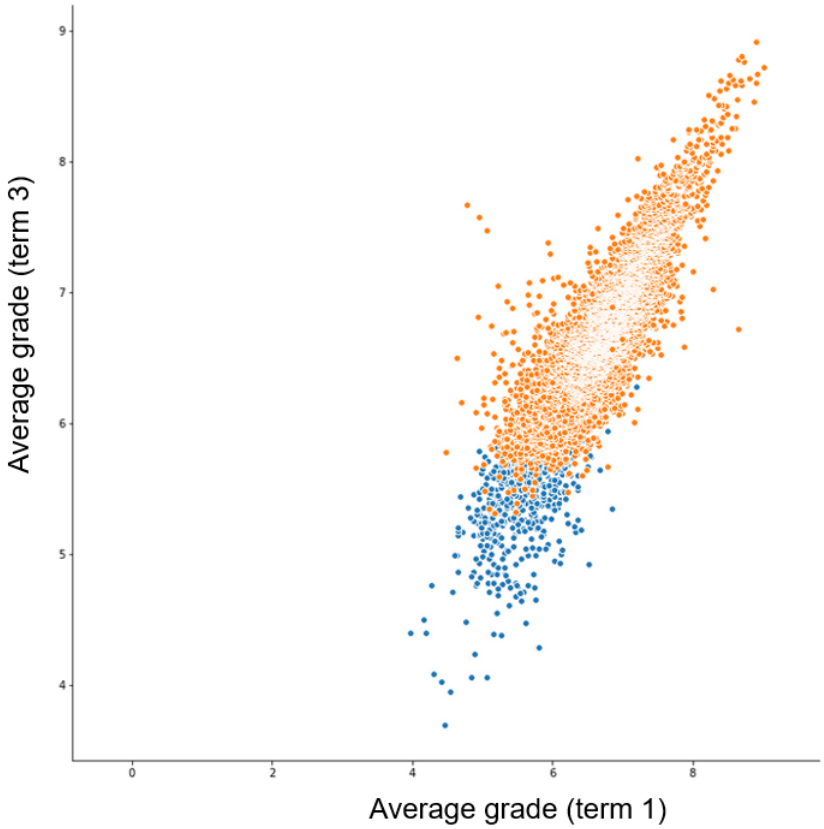

Figure 2.1: Average grades of the first and last term, indicating success at the end of the schoolyear (orange:successful, blue:not successful)

Obviously, student grades early on in the schoolyear contain some *predictive value* for student success. Certain characteristics of these grades (like the mean, number of inadequate grades etc.) may be used in a model to predict individual success and student success ratio.

This potential motivates the option of using *Machine Learning* for the prediction of student success. A machine learning model may not only use grades, but also intelligence tests and other data possibly containing predictive value. As a lot of this data is captured in the School Administration System (SAS), it may easily be used in machine learning models for student success prediction.

3 The Group Performance Problem

In this section, we will show that using a traditional Machine Learning model for student success based only on course marks reveals some issues that should be addressed in the research assignment. First, a certain dataset is established to use in the model. Subsequently, a logistic regression (LR) model is created to predict both individual student success and success ratio for each department. In the last subsection, an observation is identified which will be introduced as the Group Performance Problem.

3.1 Dataset

We are able to use data from Student Administration System (SAS) of the Dutch secondary school introduced in chapter 2, grouped by different student departments. The data of four schoolyears (the calendar year of transition ranging from 2014 to 2017) is used to create the models. Data of the last year (transition year 2018) is used to verify the model. This closely resembles the actual process of student success prediction at this institution: historical data is used to predict the outcome of the new schoolyear. The number of students in the dataset is shown in table 3.1.

Level	**Grade**	**Department**	**2014**	**2015**	**2016**	**2017**	**2018**
Mavo	2	M2	214	221	225	163	181
Mavo	3	M3	232	243	237	213	194
Havo	2	H2	172	189	154	164	163
Havo	3	H3	224	177	203	169	151
Havo	4	H4	229	270	202	268	277
Vwo	2	V2	129	128	116	127	100
Vwo	3	V3	125	129	127	114	122
Vwo	4	V4	126	105	108	117	108
Vwo	5	V5	120	125	107	112	116
		Total	**1571**	**1587**	**1479**	**1447**	**1412**

Table 3.1: Number of students in the dataset, for each department and transition year

The dataset contains aggregates of the student marks for all the courses followed in term 1, about three months into the schoolyear. Furthermore, end-of-schoolyear student success (promotion to the next grade at the same level) is included, which was extracted from the School Administration System. The success ratios of the different departments in this dataset is shown in table 3.2.

3.1.1 Imbalanced data

From the actual success ratios, it is clear that this is an *imbalanced* dataset. Fortunately, most of the students are promoted at the end of the schoolyear with the success ratios ranging from 0.80 to 0.97.
Model corrections for dealing with this imbalance (like under- or oversampling) were considered but not used. As explained in section 2, the actual class label for a student is not as important: the corresponding probabilities are of greater

Department	2014	2015	2016	2017	2018
M2	0.97	0.92	0.92	0.90	0.92
M3	0.88	0.87	0.95	0.88	0.95
H2	0.88	0.92	0.93	0.87	0.90
H3	0.87	0.80	0.87	0.92	0.91
H4	0.90	0.90	0.82	0.81	0.88
V2	0.95	0.92	0.92	0.97	0.91
V3	0.91	0.87	0.89	0.85	0.90
V4	0.92	0.90	0.87	0.88	0.81
V5	0.92	0.90	0.92	0.93	0.87

Table 3.2: Actual success ratios of students in the dataset, for each department and transition year

value. The goal is not to improve classification accuracy, but to provide reliable probabilities for student success. Note that the full set of students in the departments is used for training the models, not a sample. Correcting for the imbalance would violate the actual success chances.

3.2 Logistic Regression model using student grades

A logistic regression model is created for each department and fitted with the department-specific dataset, including the training target (student success at the end of the schoolyear).
A Logistic Regression model provides probabilities, in this case success chances, that are well calibrated [Niculescu-Mizil and Caruana, 2005]. The probabilities are easily aggregated to establish success ratios for a department.

3.2.1 Features

Some characteristics of the student grades are used as features. Note that students receive marks for *courses*, while in most departments the courses followed are different for each student, so we have to use some characteristics of the set of course marks. For this model, the following aggregates are used as features.

- **Average mark**
  Average of the course marks in the term. At this school, course marks are expressed as decimal numbers on a rating out of 10.
- **Number of shortage points**
  Total number of shortage points for all course marks in the term. The (rounded) rating of 6 out of 10 is considered satisfactory. A rounded rating of 5 out of 10 corresponds to one shortage point, a rounded rating of 4 out of 10 to two shortage points etc.
- **Minimum mark**
  The minimum course mark in the term.
- **Number of shortage points in core sections**
  Total number of shortage points within the core sections. Core sections are English, Dutch and mathematics.

These features are chosen as all of these characteristics are an important part of the *passing standard* at this school. Most of them are based on the passing standard of national exams for secondary education in the Netherlands. Some other characteristics included in the dataset are not used in the model, as they are either irrelevant for passing or correlate highly with the features above.

3.2.2 Results

The model for each department and term is fitted to the training set of students in the transition years 2014 to 2017. Subsequently, the model is tested using the data of the transition year 2018. In table 3.3, the predicted and actual success ratios are shown.

Dept.	Predicted	Actually
H2	0.79	0.90
H3	0.82	0.91
H4	0.88	0.88
M2	0.90	0.92
M3	0.90	0.95
V2	0.88	0.91
V3	0.87	0.90
V4	0.81	0.81
V5	0.88	0.87

Table 3.3: Predicted (term 1) and actual student success ratios (Logistic Regression model, training set 2014-2017, test set 2018)

The performance of this model based on student grades varies. Apparently after term 1, success prediction early on in the schoolyear, the success ratios of the departments *H2* and *H3* are severely underestimated. Surely, there must be something wrong here. Note that the fact of students improving over the schoolyear (section 2) is already captured in this model, as term 1 grades and actual end-of-year success are used for both training and test set of the model. However, the model still underestimates student success probabilities. We will introduce this issue as the *Group Performance Problem*.

3.3 The Group Performance Problem

Why is the success ratio prediction in term 1 of departments *H2* and *H3* that low? Considering the historical success ratios for these departments (table 3.2), the prediction of 0.79 for department *H2* in particular seems off. However, note that the model is based only on the course marks of students in the term.
If the performance (course marks) of the new generation (current schoolyear) is significantly lower than the performance of the students in the training set (historical schoolyears), a machine learning model obviously predicts a lower success ratio. The student results in term 1 within these departments in 2018 must generally be lower than the historical results (2014-2017). To verify this observation, the corresponding probability density functions of the average grades in both groups are shown in figure 3.1.

Clearly, the marks of the population in 2018 have become worse compared to the training population (2014-2017). From domain knowledge, we know there may be several reasons for this kind of shift. It is unlikely that a student cohort performs poor compared to earlier cohorts by chance. Usually, these shifts have to do with changing department visions and examination systems. Indeed, a change in the examination system was introduced at this school in the schoolyear 2017-2018, and students in departments *H2* and *H3* turned out to be significantly affected by this change.

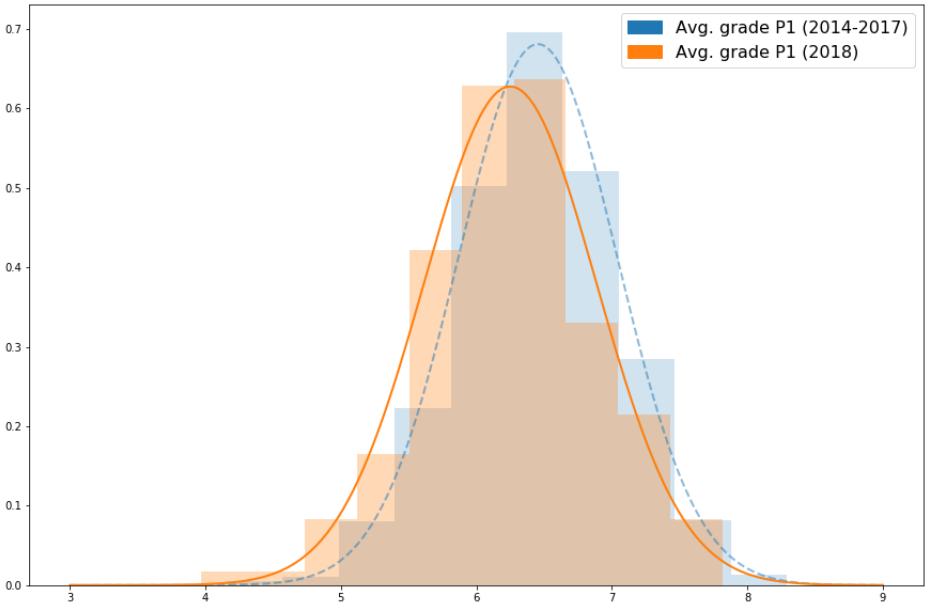

Figure 3.1: Probability Density Functions of average course marks in term 1, department H2 and H3 (orange:2018, blue:2014-2017)

Student marks being significantly lower than usual, one should expect a larger number of students failing at the end of the schoolyear. However, this is not the case: the actual success ratio of these departments in 2018 was 0.90 and 0.91, respectively (table 3.3). From domain knowledge, there are three important reasons for this difference.

- **The number and intensity of student interventions is increased**
  Disappointing results for a department after the first term are noticed by the department and school leadership. Consequently, the number of student interventions and the intensity of these interventions is increased. As secondary schools in the Netherlands should meet certain standards set by the government for promoting students [Onderwijsinspectie, 2018], resources for student interventions are increased.

- **Teachers are triggered to produce better results**
  Teachers of the affected department are triggered to improve results of their students. Consequently, they may increase the preparation of students for future examinations, or even (controversially) decide to decrease

the difficulty of tests.

- **The passing meeting promotes more students not meeting the passing standard**
  At the end-of-schoolyear passing meeting, consisting of the department teachers and leadership, there is a trigger to promote more students that do not actually meet the passing standards. There may even be *overcompensation*, explaining the high success ratio of the departments concerned in 2018 despite worsened student performance.

Clearly, these kind of shifts, and the corresponding corrections, are to be expected again in the future. Apparently, raw student mark aggregates do not provide enough information for a reliable prediction model. To improve student success predictions, the *Group Performance Problem* described in this subsection should be addressed accordingly.
As this problem is not exclusive to the domain of student success prediction, we will formulate a more general definition.

Definition

The **Group Performance Problem** (GPP) in Machine Learning may arise if predictive models are constructed for a new group of data, based on training data of one or more other groups, where both data sets are generated from the same data generating mechanism. If the probability distribution of the features in the new group significantly differs from the probability distribution of the features in the training groups, and it is known that this difference will lead to a different number and/or intensity of interventions with respect to the individuals in the new group, a traditional predictive model using these features may fail to provide a reliable prediction.

Dealing with the GPP will be the main focus of our research. The example of student success prediction will be used as a real-life case study to test the developed strategies.

4 Scientific Context

In section 2, the potential of using Machine Learning for the task of student success prediction was shown. In this section, some aspects of this topic relevant to the research project are recapitulated. Furthermore, the research question is established, related work is analysed and the scientific contribution is discussed.

4.1 Machine Learning Classification

Classification is a subset of Machine Learning where a certain record (a vector of properties, an image etc.) is assigned to one of two or more *classes* based on assigned *patterns* belonging to these classes [Theodoridis, 2015]. These patterns are created by using models that are *trained* using characteristics of observed data called *features*. If the class of the observed records is known, the classification task is a *Supervised Learning* algorithm. Classification problems may be *binary* (e.g. there are only two possible classes, usually true and false), or *multiclass* (three or more classes).

The task of student success prediction is an example of a supervised classification problem: historical records, in this case students, contain features (like student marks) as well as the *class label* (successful/not successful). The goal of this problem is to *predict* this class label for students of the current schoolyear, given certain features belonging to these students at a certain stage of the running schoolyear. This is a binary classification problem, because there are only two classes. However, as non-successful student are placed at different departments or leave the school, a multiclass classification problem might also be formulated.

4.1.1 Probabilistic models

As shown in section 2, the actual class label for an individual student is not very useful for the goals of success (ratio) prediction. The corresponding *class probabilities* are of greater value. For example, in predicting individual success, the probabilities turned out to be useful for efficiently targeting student interventions.

4.1.2 Probabilistic classification

In the case of success ratio prediction, it is perfectly fine for an individual student prediction to be, for example, 72% successful. The expected number of successful students in a certain department, and with that the success ratio, may be easily derived from these individual predictions. Since the sum of independent Bernoulli random variables with different expectations results in a *Poisson Binomial Distribution* [Daskalakis et al., 2015], the expected number of successful students is the sum of the individual probabilities. By averaging these individual chances for all students in the department, a prediction can be made for the departments' success ratio. It might be argued that these individual probabilities are not independent, which is why there is a GPP in the first place. However, the mean of these (possibly corrected) individual success chances will still determine the success ratio for the department if the probabilities are correct.

The approach of using probabilities for the classes in Machine Learning is called *Probabilistic Classification*: we are not (only) interested in the predicted label, but mainly in the corresponding class probabilities. This goal has implications for the choice of Machine Learning methods for our research: we should be careful in the selection of algorithms used for predictions of student success. For example, a certain algorithm may be quite accurate on predicting actual classes, but fail to provide reliable probabilities for all possibilities.

4.1.3 Generative and discriminative models

In generative models, a joint distribution over the feature vector $\mathbf{x}$ and class label $\mathbf{y}$ is derived. At first, a parametric distribution for $P(X, Y)$ is assumed, and using the training data the corresponding parameters are learned. Once these parameters are known, the joint distribution allows the 'generation' of new data: the reason these kind of models are called generative. An example of a generative model in the context of probabilistic classification is *Naive Bayes* [Theodoridis, 2015].

Instead of deriving a joint distribution, the conditional distribution $P(Y|X)$ may also be learned directly. In discriminative models, a parametric distribution for this conditional probability is assumed. Again, the corresponding parameters are learned using the training data. With such models, we can 'discriminate' between the classes (possibly including class probabilities) for any given new data point: the reason these kind of models are called discriminative. An example in the case of probabilistic classification is *Logistic Regression* [Theodoridis, 2015].

4.1.4 Frequentist and Bayesian methods

Corresponding to a certain view on probability, two types of methods for providing *inference* in Machine Learning are distinguished: *frequentist* and *Bayesian* statistics [Xue and Titterington, 2008]. In the frequentist view, probabilities are generally related to frequencies of certain events (explaining the name). In the case of Bayesian statistics, probabilities are related to certain *beliefs*. To provide inference this way, there is an initial (*prior*) belief, which is updated as new data is observed, resulting in a *posterior* belief. With new *evidence*, the posterior becomes the new prior and a new posterior is calculated.

The main difference between the two methods is the use of a prior belief in Bayesian statistics, whereas in frequentist methods only the observed data is used to provide inference.

4.2 Dataset shift

The phenomenon of training and test data having different distributions, though the terminology is not completely consistent in scientific literature, is known as *dataset shift*. Different types of dataset shift were documented by Moreno et al. in an attempt to standardize terminology in this field [Moreno-Torres et al., 2012]. The authors also provided an overview of the existing strategies for machine learning under the situation of dataset shift.

4.2.1 Types of shift

The following types are mentioned by Moreno:

- **Covariate shift**
  In this situation the input parameters (features) differ from training to test set, but the conditional probability of the target variable Y with respect to features X is not changed, so:

$$\begin{aligned} P_{train}(X) \neq P_{test}(X) \quad and \\ P_{train}(Y|X) = P_{test}(Y|X) \end{aligned} \tag{1}$$

- **Prior probability shift**
  This type of shift appears in generative models if only the distribution of Y (the class variable) is changed, effectively covariate shift in reverse. In this case:

$$\begin{aligned} P_{train}(X|Y) = P_{test}(X|Y) \quad and \\ P_{train}(Y) \neq P_{test}(Y) \end{aligned} \tag{2}$$

- **Concept shift**
  In this type of shift the relation between features X and class variable y changes from training to test, but the probability distribution of X does not differ, I.E. in classification:

$$\begin{aligned} P_{train}(X) = P_{test}(X) \quad and \\ P_{train}(Y|X) \neq P_{test}(Y|X) \end{aligned} \tag{3}$$

The Group Performance Problem, as defined in section 3, is a combination of *covariate shift* and *concept shift*. Both the distribution of features X and the conditional probability of the class variable y given the feature variables X differ from training to test. I.E.:

$$\begin{aligned} P_{train}(X) \neq P_{test}(X) \quad and \\ P_{train}(y|X) \neq P_{test}(y|X) \end{aligned} \tag{4}$$

This situation is recognized by Moreno, with the authors stating:

"There are two main reasons these shifts are usually not considered in the literature: they appear more rarely than the others and, most importantly, they are so hard that we currently consider them impossible to solve." (p525).

The Group Performance Problem, however, is a subset of this type of dataset shift. In this case the distribution of X changes from training to test data, but the conditional probability of class variable Y given features X changes accordingly to neutralize the effect on Y to some extend (the strength of the intervention effect). This property might result in the ability to create models that are capable of dealing with the Group Performance Problem, which will be addressed in our research.

4.2.2 Concept drift

It should be noted that terminology has somewhat changed over time. In a review of Lu et al., focusing on changes in the probability distribution of streaming data, the term *concept drift* is used by the authors as an umbrella term [Lu et al., 2019], defined as:

$$\exists t : P_t(X, Y) \neq P_{t+1}(X, Y) \tag{5}$$

Like dataset shift, concept drift is defined as the change of the joint probability of features X and target variable Y over time. The different types are presented by the authors as three *sources*:

- **Source I**

$$\begin{aligned} P_t(X) &\neq P_{t+1}(X) \quad while \\ P_t(Y|X) &= P_{t+1}(Y|X) \end{aligned} \tag{6}$$

  corresponding to *covariate shift*

- **Source II**

$$\begin{aligned} P_t(Y|X) &\neq P_{t+1}(Y|X) \quad while \\ P_t(X) &= P_{t+1}(X) \end{aligned} \tag{7}$$

  corresponding to *concept shift*

- **Source III**

$$\begin{aligned} P_t(X) &\neq P_{t+1}(X) \quad and \\ P_t(Y|X) &\neq P_{t+1}(Y|X) \end{aligned} \tag{8}$$

  corresponding to a combination of *covariate shift* and *concept shift*.

Though not exactly falling within streaming data, the Group Performance Problem is a special case of source III: both the distribution of the feature variables (lower student marks) and the conditional probability of class variable Y given these features (chance of passing based on student marks) change from historical schoolyears (training set) to the current schoolyear (test set), which is obviously a change over time.
To conclude the terminology in scientific literature, the phenomenon of covariate shift (source I) is also known as *virtual drift*, and concept shift (source II) may be addressed as *real drift* [Gama et al., 2014].

4.3 Research Question

Multiple strategies for dealing with the group effect will be examined. As early student marks by themselves clearly do not provide sufficient predictive value for end-of-schoolyear success, more information should be incorporated in the models. We may be able to adjust the features in traditional models, or create probabilistic models using information about student success ratios, i.e. using Bayesian methods.

Consequently, we will use both frequentist and Bayesian methods in the modelling part of the project. Therefore, the research question for this project is as follows:

RQ **Research Question**
How can we successfully improve Student Success Prediction by dealing with the Group Performance Problem using (Bayesian) Machine Learning?

To be able to answer the research question, different aspects should be examined. First, we should know the performance of models based only on student course marks (part of which is already presented in section 3 using a traditional Logistic Regression model). To improve this performance, different strategies to deal with the Group Performance Problem are distinguished: incorporating more and/or adjusted features in the models, and using group performance information in a Bayesian model. The corresponding sub questions are as follows.

SQ1 **Sub Question 1**
How does traditional Student Success Prediction using Machine Learning based on early student course marks perform?

SQ2 **Sub Question 2**
How can we improve this prediction using *additional* features?

SQ3 **Sub Question 3**
How can we improve this prediction by *adjusting* the features?

SQ4 **Sub Question 4**
How can we improve this prediction with probabilistic models using Bayesian methods?

SQ1 is partly answered by the analysis in section 3. We will expand this analysis with other classification algorithms, like the *Support Vector Machine* and *Tree Based Models*. The GPP is expected to appear in all these models, if absolute student mark aggregates are used as features.

In SQ2 and SQ3 respectively, these classification models are expanded with more or adjusted features. Additional information like student intelligence test scores is incorporated in training and test data to provide learning possibilities that go beyond student grades alone. Furthermore, features may be adjusted to create models that are less sensitive to the group effect. For example, deviations from the mean group mark features may be used instead of actual values.

SQ4 presents the opportunity to create a different kind of model for evaluating student success probabilities. Using Bayesian methods, more information may be incorporated in the model, decreasing the impact of student course marks on the class probabilities. Note that not only historical student marks in term 1 and whether the students were successful at the end of the schoolyear are known, but also the historical group success ratios as shown in table 3.2. Using this information in a probabilistic model may be a successful strategy in dealing with the GPP.

4.4 Related work

Extensive research has been performed on *concept drift*, dataset shift with respect to streaming data as mentioned in subsection 4.2.2. In a recent review by Lu et al., three groups of methods adapting to concept drift are identified [Lu et al., 2019]:

- **Retraining**
  In this case, a new model is trained to be used instead of the model based on training data with a different distribution. The new model is generally trained with only the more recent data, incorporating the shift in distribution. An example of this method was used by Bach et al. using both a *stable* learner using all historical data and a *reactive* learner with only recent data for training, followed by detecting and adapting to a possible shift in the distribution [Bach and Maloof, 2008].

- **Ensemble training**
  Ensemble training combines a set of different classifiers, and uses some sort of voting system for the actual classification of new data. To address concept drift, extended ensemble methods have been developed. For example, Bifet et al. created *Leveraging Bagging* where the worst performing classifier in the ensemble based on recent data is replaced by a new classifier based on this new distribution [Bifet et al., 2010].

- **Model adjusting**
  If the change in data distribution is mostly local, existing models may adjust to this change. As only part of the model needs to be retrained, this method is mostly used in decision tree models. Hulten et al. developed the algorithm CVFDT, an extension of the fast decision tree learner VFDT [Hulten et al., 2001]. In the model, the most recent data is captured in a moving window, on which alternative sub-trees are trained. Both the performance of the original and alternative sub-trees is monitored, and older sub-trees are removed and replaced by alternatives if they are outperformed.

It appears that, with the enormous growth of *big data* analysis, most research on dataset shift is now focused on streaming data. The task of student success prediction is not totally different from a streaming scenario: a prediction is made every schoolyear, and the amount of training and test data keeps growing in the future. However, the Group Performance Problem introduces some unique characteristics that should be addressed by our research.

4.5 Scientific Contribution

All currently available strategies for dealing with dataset shift, as shown in the preceding subsection, are based on some sort of detection of a change in the distribution of the data, and subsequently adapt to this change by retraining or adjusting models. To detect the shift however, there is an important limiting factor as noted by Lu et al. in the concluding remarks [Lu et al., 2019]:

"Most existing drift detection and adaptation algorithms assume the ground true label is available after classification/prediction, or extreme verification latency" (p2359).

Note that in student success prediction, the prediction needs to be adapted *before* the actual class labels of the new group are known. The distribution change of only the predicting features, and the domain knowledge of increased interventions as a reaction on this change, is enough for requiring a change in the model. Furthermore, successful models dealing with the group effect should not only address the problem if a comparable shift has appeared earlier in the training data. The occurrence of the problem within a department is too low to assume such a historical shift, which might hold for other domains dealing with a comparable problem.

So obviously, there is room for a solid scientific contribution of our research. In the Group Performance Problem, models should adapt even before the true class label of new data is known, and a distribution change is detected. The problem is also unique because of the group-wise streaming data: intervention effects depending on the group performance, the aggregate of the individual predictions within the group (predicted success ratio). Fortunately, this domain knowledge makes it possible to detect and predict a dataset shift before the actual class labels are known. Strategies for dealing with this specific situation will be a clear contribution to the scientific literature in the field of dataset shift and adaptive learning.

5 Methods

Different strategies for dealing with the GPP will be developed and tested, corresponding to the research sub questions. In this section, we will describe the methods used for this empirical part of the research project.

The models will be compared using a synthetic dataset, containing a certain degree of the group effect, which will be quantified in this section. Furthermore, the simulation process of generating synthetic data will be discussed. Finally, the developed models are presented, and we will establish an objective way of judging the performance of the different models, both in the synthetic data scenario and the student success prediction case study introduced in section 3.

5.1 Synthetic data

A set of synthetic data is generated with a varying strength of the group effect. This corresponds to reality, as the number and intensity of interventions in response to an observed feature distribution change may vary in different scenarios. The models are tested with all these variances to analyse their performance in all situations.

5.1.1 The Group Performance Factor

To quantify the strength of the group effect, we need to realize that this phenomenon actually consists of two parts:

- **A change in the distribution of covariates (features)**
  The distribution change is noticed by comparing it to the historical feature distributions. The change may be reviewed depending on the variance of these historical distributions.
  If there is a low variance in the feature aggregates of these historical groups, the change will be reviewed differently from scenarios where changes in the distribution are frequently observed. For example, if student course mark distributions in term 1 vary substantially from year to year, the distribution for the new group may not be evaluated as problematic. However, if the historical course mark distribution is rather constant, the change may well be noticed and reviewed by the department leadership.

- **An increased number and/or intensity of interventions**
  Interventions may or may not be increased with the observed feature distribution change. If the interventions do not differ from those in historical groups, traditional models will probably provide an acceptable prediction (though the predictive model may be adjusted to account for a *covariate shift*).
  Only in the case of increased intervention effects, the conditional probability of class variable Y given features X changes. Furthermore, the effect of interventions may obviously vary in strength.

These two parts should be reflected in a factor quantifying the strength of the GPE. The first part (change in distribution) is easily evaluated at the time of prediction, as all historical and current feature information is available.

However, the second part (intervention effect) is not known at that point, and should be evaluated by domain experts. After the true class labels of the individuals in the new group are available, this effect may be determined by comparing the true success ratio to the predicted ratio of a base model without correction for group effects, like the standard Logistic Regression model. The difference between these ratios is a measure for the strength of the intervention effect (both the number and/or intensity of interventions and their effect on the true class labels).

In view of these considerations, we propose the following factor to determine the strength of the Group Performance Effect:

Definition

In a Machine Learning scenario where the Group Performance Problem is applicable, the **Group Performance Factor** (GPF) for the binary classification of a new group of individuals, is defined as:

$$\mid \frac{\mu_{\overline{y}_{tg}} - \overline{y}_{base}}{\sigma_{\overline{y}_{tg}}} \mid (\overline{y}_{test} - \overline{y}_{base}) \tag{9}$$

Where:

- $\overline{y}_{tg}$ is the ratio of positive class labels in each training group (the *group success ratio*)
- $\mu_{\overline{y}_{tg}}$ is the mean of this ratio for all training groups in the dataset
- $\sigma_{\overline{y}_{tg}}$ is the standard deviation of the group success ratio for all training groups in the dataset
- $\overline{y}_{base}$ is the success ratio prediction for the new group by a *base* model (like an unmodified Logistic Regression model)
- $\overline{y}_{test}$ is the actual success ratio for the new group

The first term of equation 9, which we will call the **group shift**, indicates the shift of the base prediction compared to the historical group success ratios, i.e., how many standard deviations the base prediction differs from the mean of historical group success ratios (this is similar to the Z-score known in statistics). Note that the feature distribution change is modelled implicitly by using a (naive) base model to create a prediction for the new group. This method prevents a cumbersome comparison between the feature distributions typically consisting of a large number of variables of different types.
Dividing by the standard deviation of the group success ratios in the training data makes sure that the GPF is higher if the historical group success ratios are rather constant, and the GPF is lower if there is already a history of fluctuating success ratios.

The second term indicates the **intervention effect**, the difference between the actual success ratio and the one predicted by the (naive) base model. If interventions were not different in number or intensity compared to those in historical groups, the actual success ratio will not be significantly different from the predicted ratio and this term will be close to zero. However, in the case

of intensified interventions, the difference may be quite substantial, resulting in a higher GPF. Note that this term may take negative values, as success ratios may also be restrained by interventions in certain scenarios or other domains.

From both terms, we may conclude that a high GPF will result if both the prediction of a base model differs from historical groups (group shift) and the number or intensity of interventions is increased as a reaction on this change (intervention effect). If one of these terms is close to zero, we either have a distribution change without a significant change in interventions (pure covariate shift), or a changing relation between the features and actual class label (pure concept shift), for which other solutions may apply.

Note that the intervention effect is not known by the time of prediction. Only after the actual class labels for the new group are established, the second term of equation 9 may be determined. In a real life scenario, the intervention effect would have to be evaluated by domain experts by determining the increase of interventions and estimating their effects on the actual class labels in the new group.

5.1.2 Data simulations

To compare the models that will be developed, synthetic data is generated with an increasing GPF. This enables us to evaluate the models with varying strengths of the group effect, and possibly suggest the use of certain models in different situations.
The case of student success is simulated in a simplified scenario. Only one predicting covariate (feature) is generated, the mean course mark. Groups of 200 students are simulated, with five training groups and a test group (the new group for which a prediction should be made). The steps in this simulation process are as follows.

- Mean course marks for the individuals in the five training groups are drawn from a normal distribution with mean 6.2 and standard deviation of 0.7 (these values actually correspond to those in the full dataset of our example school).
- Mean course marks for the test group are drawn from a normal distribution with mean 5.8 and the same standard deviation of 0.7
- For the training groups, an improvement of this feature is randomly drawn from a normal distribution with mean 0.1 and standard deviation of 0.4 (again, these values correspond to actual course mark improvements over the schoolyear in our case study data)
- For the test group, this improvement is randomly drawn from a normal distribution with varying mean (ranging from 0.1 to 0.55) and the same standard deviation as the improvement in the training groups
- The actual class label is determined with success boundary 6.3: if the mean course mark with the improvement exceeds this boundary, the individual is assigned a positive class label (success), otherwise a negative one (not successful)

- For each varying improvement of the test group (intervention effect), the simulation is repeated 200 times with results averaged to reduce variance

The success boundary (cut-off point) was iteratively determined to result in more or less balanced datasets in the training groups. Note that in reality, the success boundary is closer to a mean course mark of 5.5, resulting in the imbalanced case study dataset as shown in section 3.1. In this simulation process however, we should avoid introducing certain unknown effects that may result from an imbalance in the generated data.

The models should provide a prediction for all the cases generated (all variations of test group improvements, repeated 200 times). For each iteration, the Group Performance Factor is calculated and the model performance is noted. This results in a dataset of model results for an increasing group effect, which eventually enables us to provide recommendations about the use of certain models in varying strengths of the GPF.

5.2 Case Study: School dataset

All models will also be tested with the real dataset from our example school regarding student success prediction. The occurrence of the GPP in this dataset was already shown in section 3, particularly in department *H2*. Now that we have defined the Group Performance Factor, we may calculate this value for the different departments in the dataset.

With training groups for transition years 2014, 2015, 2016 and 2017, the GPF for the new groups (transition year 2018), is shown in table 5.1.

Level	**Grade**	**Department**	**GPF**
Mavo	2	M2	0.01
Mavo	3	M3	0.01
Havo	2	H2	0.38
Havo	3	H3	0.08
Havo	4	H4	0.00
Vwo	2	V2	0.05
Vwo	3	V3	0.00
Vwo	4	V4	-0.02
Vwo	5	V5	-0.05

Table 5.1: Group Performance Factor for different departments (training set 2014-1017, test set 2018)

The high GPF of 0.38 for department *H2* clearly stands out, as expected. For this department in 2018, both the *group shift* and the *intervention effect*, the left and right terms of equation 9 respectively, are considerable. The GPF for department *H3* is also high compared to other departments, but does not come close to that of department *H2*.

In the model analysis with respect to the case study, we will evaluate two results:

- **Model performance for department H2 (2018)**
  The models are evaluated for this department in transition year 2018, with the data from schoolyears 2014-2017 as training set. As this department clearly shows a considerable GPF, the performance with respect to this department of each model that will be developed is of high interest.
- **Model performance for all departments (2018)**
  Models that are able to deal well with a high GPF, should preferably not perform less when the group effect is relatively weak. Therefore, we will analyse the models for the whole set of departments in transition year 2018. As the majority of departments show a GPF close to zero (table 5.1), this analysis will be interesting to evaluate the general performance of models, not only in the case of a high GPF.

5.3 Model performance: Brier score

As shown in section 2, the actual predicted class labels for a new group of students within a department is of little interest. The individual success probabilities, the predicted probability of a positive class label in the classification problem, is of greater value. Consequently, traditional classification metrics like *accuracy* or *precision/recall* are not suitable for this particular problem.

To score a probabilistic forecast with respect to the actual outcome, different metrics may be used. These *scoring rules* may be formally defined as *proper* or even *strictly proper* [Merkle and Steyvers, 2013]. A scoring rule is proper if the score function is minimized in the case of the forecast approaching the actual probability. This is the case for a lot of scoring metrics, so a subset may be defined: strictly proper scoring rules. In this case, the metric is minimized if *and only if* the forecast equals the actual probability (or outcome for an infinite number of experiments).

As we would like our models to predict probabilities that correspond to the actual probability as much as possible, the strictly proper *Brier score* [Brier, 1950] is used to evaluate the prediction of the individuals in the new group. The Brier score is a quadratic score metric indicating the squared error between predicted probability and actual outcome. Obviously, the goal of developing our models is to realize a Brier score as low as possible, even when the Group Performance Factor is high.

Note that we already created a (base) model in section 3. We may now expand the results of table 3.3 with the Brier scores of this model for the different departments. These scores, or mean squared errors, are shown in table 5.2.
The Brier score of this model for the department with a notably high GPF (department *H2*) is the second-highest in the set. Department *H3* also shows a considerable error with this unmodified Logistic Regression model compared to other departments. This is expected, as the presence of a GPE results in an inferior forecast with a traditional model by definition.
Note, however, that a high Brier score (lower performance) may not result from a group effect exclusively. In department *H4* and *V4* the GPF is small (table 5.1), but the Brier scores are high compared to other departments. It may be

Dept.	Predicted	Actual	Brier score
H2	0.79	0.90	0.076
H3	0.82	0.91	0.073
H4	0.88	0.88	0.067
M2	0.90	0.92	0.042
M3	0.90	0.95	0.038
V2	0.88	0.91	0.053
V3	0.87	0.90	0.037
V4	0.81	0.81	0.080
V5	0.88	0.87	0.045

Table 5.2: Predicted (term 1) and actual student success ratios, including Brier score (Logistic Regression model, training set 2014-2017, test set 2018)

concluded that though the *success ratio* is predicted well for these departments, predicted individual probabilities are off. This is not the result of a group effect affecting all individuals, but simply due to the fact of part of the students developing unexpectedly during the schoolyear (e.g. students with low course marks early on actually being promoted at the end of the schoolyear, and the other way around).

The Brier score will be used as the single metric for assessing the different models, both in the simulations with synthetic data and in the case study of student success prediction. We aim to develop models that are able to perform well (lower Brier score) in classification problems with different Group Performance Factors, or at least provide some recommendations for when to use which model depending on the strength of the group effect.

6 Models and implementation

Different strategies for dealing with the Group Performance Problem (GPF) are developed, corresponding to the research sub questions (4.3). In this section, the corresponding models are discussed.

6.1 Traditional Machine Learning models (SQ1)

In the first sub question, the performance of student success prediction using traditional Machine Learning (ML) models is examined. Obviously, to determine improvement using the strategies from the other sub questions, we should know the prediction quality of unmodified models. Different types of ML models are examined to have a broad perspective of the performance of a traditional approach dealing with the GPP.

An example of a traditional model was already shown in section 3, where a Logistic Regression model was developed to illustrate the Group Performance Problem. We will expand that analysis with some other regularly used ML models. For implementing the frequentist models and using them in the analysis, the Python library *sklearn* is used [Pedregosa et al., 2011]. Since we are mainly interested in predicting individual student success probabilities (see section 2), the models should provide true posterior probabilities. For most models, *sklearn* provides the *predict_proba* method, resulting in a probability of the target class instead of a binary classification.

Not all learning algorithms result in well calibrated probabilities naturally, and predictions may be distorted [Niculescu-Mizil and Caruana, 2005]. If this is the case, a correction called *Platt Scaling* may be used [Platt, 1999]. In this method, outputs are projected to a sigmoid function resulting in posterior probabilities. This property makes sure that all models are able to provide a calibrated success probability, which may be used to determine the performance of our models. Fortunately, Platt Scaling is integrated in the *predict_proba* method of *sklearn* if applicable.

The following ML models were created without significant modifications:

- **Logistic Regression**
  In a Logistic Regression (LR) model, the logistic function is fitted to provide the probability of a certain input vector (features) belonging to the positive class. In our case, we are mainly interested in this predicted probability, but LR may obviously be turned into a binary classifier by selecting a *cut-off* value for the probability (usually 0.5).
- **Support Vector Machine**
  A Support Vector Machine (SVM) creates a *hyperplane* that separates the two classes by the maximum margin. Consequently, training data points closer to the *decision boundary* have more influence on the separating hyperplane. The classifier is not probabilistic by nature, so *Platt Scaling* is used to output (calibrated) probabilities.

- **Random Forest**
  The Random Forest (RF) classifier is an *ensemble* method that creates multiple *decision trees* (hence the name 'forest'). The ensemble generally prevents overfitting, which is a serious risk when using single decision trees. As decision trees only provide a binary outcome, to return a probability *sklearn* computes the fraction of the trees in the forest predicting the positive class label.

It should be noted that, in all models, the input feature vector is transformed by using sklearns' *StandardScaler*, which essentially removes the mean of the features and scales them to unit variance. Some models (like the SVM with a non-linear kernel) assume this property, and in other cases it is assumed that the transformation will not significantly affect model performance.

6.2 Adding more features (SQ2)

A straightforward approach to improve Machine Learning classification tasks is to include more information in the model in the form of additional features. Consequently, the features with respect to the student course marks will be less important for the prediction of the positive class probability (success chance). For the case study, our example school, the following additional features will be introduced to include additional information:

- **Recommendation PE**
  In the Netherlands, schools for *Primary Education (PE)* provide a level recommendation for each pupil after eight years of education. This level recommendation is known in the administration system of secondary schools, and may be used to determine the best individual placement of students during the early grades. The recommendation may be expressed as a number between 1 and 6 (corresponding to the different levels in secondary education).

- **Intelligence and skill score (Mathematics)**
  The CITO skill score with respect to Mathematics. During the first three grades, students in our example school are independently tested by Dutch institution CITO. These tests do not influence the course marks of the pupils, but may be used to identify flaws in the cognitive development of the students.

- **Intelligence and skill score (Dutch)**
  The CITO skill score with respect to Dutch language and receptive reading.

- **Intelligence and skill score (English)**
  The CITO skill score in the field of English language, one of the core sections as discussed in section 3.

These features are relatively independent on the course marks of the students, they provide more or less *a priori* information for the success chances of the individual students within a department as these scores are known before the schoolyear is even started. Consequently, by including these features in addition to course mark aggregates, we expect models to suffer less from the group effect

as course marks play a smaller role in the predictive model.

This strategy is only tested in the case study. Adding one or more features in the synthetic data simulations containing strong correlation with the actual class label will obviously improve model performance. However, in this case we are interested in this specific domain: do other features not specifically linked to student course marks significantly improve the prediction, or do they turn out to be redundant? If adding these features actually improve model performance, we would clearly add them to any predictive model as they appear to contain additional predictive information in that case.

6.3 Using relative course mark features (SQ3)

In Machine Learning classification, feature vectors are usually standardized by removing the mean and scaling to unit variance, the process we implemented in the models using the *Standardscaler* object provided by *sklearn*. This transformation generally improves model performance and prevents some modelling issues that may arise if the covariates are highly varying in range. In these cases, the same transformation that was established with the training data is applied to the test set.

In the strategy corresponding to SQ3, however, we deliberately use a different transformation for the new group of students (test set). A new transformation is applied to the covariates in this group, using the mean of the new group instead of the training set mean for continuous features. With this adjustment, we effectively use course mark aggregates *relative to the own group* instead of actual values. The idea behind this strategy is the assumption that, as variance in department success ratio is not too high (table 2.1), it is apparently more important how a student performs compared to the other individuals in the group than the absolute performance according to the course marks. If the course marks for all students in the new group are generally worse compared to those in historical schoolyears, success chances of students with marginal results are actually increased.

The implementation of this model is pretty straightforward, as shown in the following code block for removing the mean of the covariates.

```
def build_solution(self, X_train, y_train):
    #standardize features
    X_train_transformed = X_train-X_train.mean()
    ...

def evaluate_testset(self, X_test):
    ...
    #standardize features with respect to the new group
    X_test_transformed = X_test-X_test.mean()
    ...
```

We expect this model to perform well when there is a significant shift in the mean of the features of the test set compared to the training set, as the standardizing

process essentially means this shift has no effect on the model. However, if this difference is present but the *intervention effect* is close to zero (i.e. the results of the new group are disappointing but interventions are not increased), this model may incorrectly increase student success chances resulting in higher Brier scores and therefore lower model performance compared to a base model.

6.4 Bayesian models (SQ4)

In the Bayesian models, we will approach the problem probabilistically. A representation of the student success domain is represented in the *Bayesian network* of figure 6.1. Bayesian networks are directed graphical models, where the nodes are variables and the arcs represent the connections (usually causal relations) between these variables [Korb, 2011].

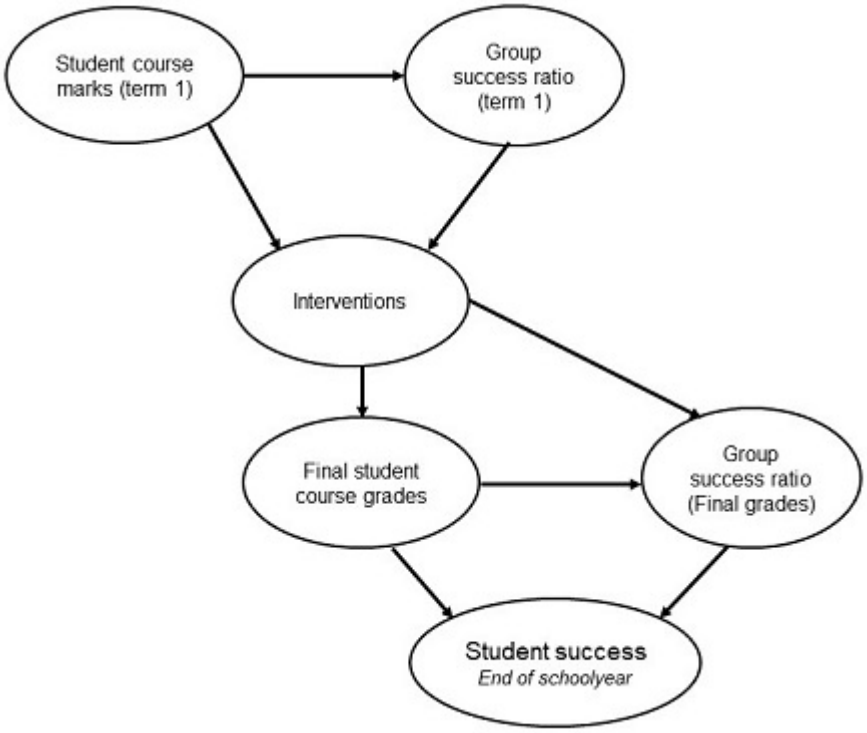

Figure 6.1: Bayesian network of the student success domain

In the last node, the student success is determined by the *passing meeting*. Note that the group success ratio based on the final grades at that point affect the decisions: if the interventions did not improve results satisfactorily, more students that do not meet the final passing standard may be promoted nonetheless.

6.4.1 MCMC and Pymc3

In our Bayesian models, we will establish *prior* probability distributions for the variables and coefficients involved. With the *evidence* provided by the training data, the goal is to provide the *posterior* probability distribution of this

set of random variables, a method called *Bayesian inference*. For non-trivial models, this process is intractable as it involves computing complex integrals [Lee and Wagenmakers, 2014].

Fortunately, there are several ways to approximate the posterior distribution. We will use *sampling* algorithms using Monte Carlo Markov Chain (MCMC) techniques. In this process, samples are generated from the posterior. If there are enough samples available, any statistic of the posterior distribution may be calculated (like the mean of a variable). Estimating the distribution with random samples is the *Monte Carlo* aspect of MCMC. The *Markov Chain* part of the sequential process of drawing samples means that samples do depend on the previous sample, but not on the samples before that one (the Markov property) [Lee and Wagenmakers, 2014].

Formally, Bayes rule states that the probability of hypothesis h, given evidence e, is equal to the *likelihood* $\mathrm{P}(e|h)$ times its prior probability $\mathrm{P}(h)$, normalized over the global probability of the evidence e [Korb, 2011]:

$$P(h|e) = \frac{P(e|h)P(h)}{P(e)} \tag{10}$$

If the likelihood is available in analytical form, the posterior may be calculated using the prior and this analytical expression. In practice, however, this expression is not available and sampling from the posterior is an acceptable alternative by approximating the posterior distribution.

For our models, we will use the Python library *Pymc3* [Salvatier et al., 2016]. This library provides a framework for defining models without a domain-specific language and offers multiple MCMC samplers. The No-U-Turn Sampler (*NUTS*) is used in all our Bayesian models for the sampling process. NUTS is a Hamiltonian Monte Carlo (HMC) algorithm that converges faster than unmodified random walk samplers like *Metropolis* or *Gibbs* sampling [Homan and Gelman, 2014]. The major advantage of this sampler is the auto-tuning aspect, enabling us to avoid an intensive manual tuning process. In practice, we found that tuning only the step size of the NUTS algorithm (using the *target_accept* parameter) is sufficient to establish a converging sampling process in all models.

Since we are mainly interested in the mean of certain variables to create a predictive model, 2000 samples for each model and simulation are used for approximating these values (after dismissing 500 samples used for tuning the model). This number of samples turned out to be satisfactory in predicting the mean values of unknown random variables used for the predictive models.

6.4.2 Bayesian Logistic Regression

For our first predictive Bayesian model, we would like to establish the conditional probability $P(Y|X)$ directly, without including the group performance. In this case, a Bayesian Logistic Regression model is created. Like the frequentist approach of the base LR model developed for SQ1, parameters are learned to fit the logistic function:

$$\theta(z) = \frac{1}{1 + e^{-z}}$$

with

$$z = \beta_0 + x1\beta_1 + .. + x_n\beta_n$$

and n is the number of covariates (features) in the dataset. In this problem, the coefficients in parameter vector $\boldsymbol{\beta}$ are learned. The result of the function $\theta(z)$ represents the probability of a positive class label, in our case the student success probability.

The model is created using Pymc3, with weak (uninformative) prior distributions for parameters $\boldsymbol{\beta}$. The likelihood is modelled using a Bernoulli distribution with $p = \theta$ and observed values for the class labels (1 or 0 for student success) in the training data. The outline of this model, with i students in the training data containing n features, is shown in figure 6.2.

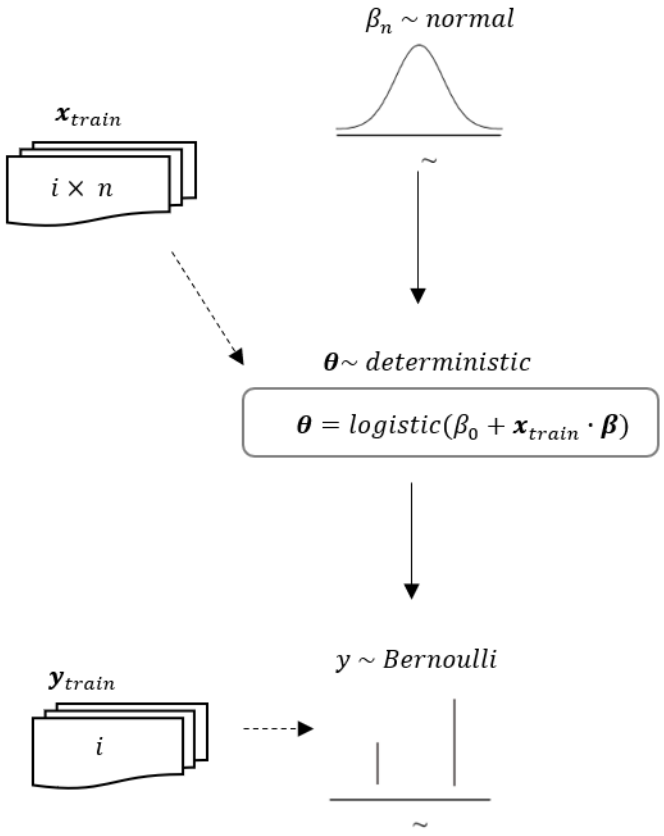

Figure 6.2: Implementation of the Bayesian Logistic Regression model

In this model, the following parameters are incorporated:

- $\boldsymbol{\beta}$
  Coefficient vector for the logistic function. Weak prior values are assumed: the prior for each β_n is modelled as a normally distributed variable with mean 0 and standard deviation 10.

- $\boldsymbol{\theta}$
  Deterministic vector of the sigmoid function applied to

  $$z = \beta_0 + \beta_1 x_1 + .. + \beta_n x_n$$

  for each feature vector $\boldsymbol{x_i}$ (student) in de training dataset. These values represent the success probability of each student ($0 \leq \theta_i \leq 1$).

- $\boldsymbol{y}$
  Bernoulli distribution with $p = \theta$ and observed values for the training dataset: the actual class label for the student (1 for success, else 0).

In the NUTS sampling process, the mean values of coefficients $\boldsymbol{\beta}$ are established, resulting in our predictive model for the new group of students. To calculate the success probability of a student in the new group, we may simply calculate $\theta(j) = logistic(\beta_0 + \beta_1 x1 + .. + \beta_n x_n)$ for each student feature vector $\boldsymbol{x_j}$ in the new group using the mean value for each coefficient β_n determined by the sampling process.
Note that, as we are using weak priors, this model is basically equal to the deterministic LR approach of SQ1. Therefore, we expect this model to perform about the same as that model. At least, by comparing the results of the equivalent models we are able to confirm the correctness of the design of our Bayesian models.

6.4.3 Bayesian Group model

The first real strategy for dealing with the group effect in a Bayesian model is incorporating the historical success ratios of former groups of students in the prediction. In this case, not only the course grades are used in the predictive model, but also the results of the group as a whole (predicted success ratio). For this group result, fortunately, we have some *evidence* for the distribution of this success ratio from year to year.

Like in the former model, a Bayesian Logistic Regression model is created. The class labels for the training data are still observed, but additional evidence is added for the success ratio of groups of students (departments in a certain schoolyear). The success ratio distribution is modelled based on the $\boldsymbol{\beta}$ parameter values and input features, while the historical success ratios are used as evidence for this distribution (observed variable in Pymc3).

An outline of the model is shown in figure 6.3. In this model, there are i students in the training dataset, divided in k groups (historical schoolyears). For each student, n predicting features are available. There are j students in the new group (test set), for which a prediction should be made.

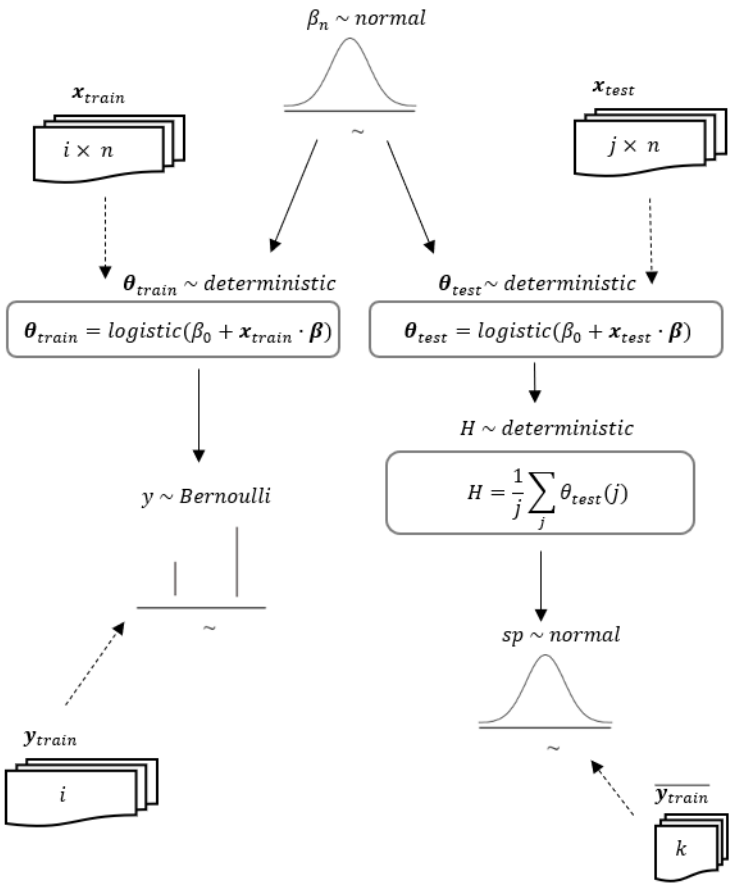

Figure 6.3: Implementation of the Bayesian Group Model

In this model, the following parameters are incorporated:

- $\boldsymbol{\beta}$
  Coefficient vector for the logistic function. Weak prior values are assumed: the prior for each β_n is modelled as a normally distributed variable with mean 0 and standard deviation 10.

- $\boldsymbol{\theta_{train}}$
  Deterministic vector of the sigmoid function applied to

  $$z = \beta_0 + \beta_1 x_1 + .. + \beta_n x_n$$

  for each feature vector $\boldsymbol{x_i}$ (student) in de training dataset. These values represent the success probability of each student ($0 \leq \theta_i \leq 1$).

- $\boldsymbol{y}$
  Bernoulli distribution with $p = \theta$ and observed values for the training dataset: the actual class label for the student (1 for success, else 0).

- $\boldsymbol{\theta_{test}}$
  Deterministic vector of the sigmoid function applied to

  $$z = \beta_0 + \beta_1 x_1 + .. + \beta_n x_n$$

  for each feature vector $\boldsymbol{x_j}$ (student) in de *test* dataset, e.g. the students in the new group. These values represent the success probability of each student in this group ($0 \leq \theta_j \leq 1$).

- H
  Deterministic variable representing the mean success probability of the new group of students (success ratio of this group) based on $\boldsymbol{\theta_{test}}$.

- sp
  Probability distribution of the success ratio. This distribution is modelled as a normal distribution with mean H and the historical group success ratios as observed values. The standard deviation is set equal to that of the historical group success ratios.

Like in the Bayesian Logistic Regression model, the mean values for coefficients β_n are determined from the sampling process, resulting in the predictive model for the new group of students (the test group).
We expect this model to have a restraining effect on the success ratio. If the initial success ratio for the new group of students based on the logistic fit for the training data is significantly lower than the historical success ratios, the β coefficients values will adjust accordingly depending on the observed historical success ratios. Note that this effect is stronger if the observed historical success ratios have low variance: the model will provide a stronger correction if the *group shift* as defined in the left part of equation 9 is higher.

6.4.4 Bayesian Lambda model

The second probabilistic strategy is different. In this model, the success probabilities of the students in the new group are predicted using an unmodified Logistic Regression model fitted with the training dataset. We will use the frequentist model of RQ1 (implemented with *sklearn*), but this could also be the Bayesian variant previously discussed.

In this model, however, we expect this probability to increase (or decrease in certain scenarios), based on the predicted success probability by this unmodified model and the observed historical success ratios. The increase is defined by the following equation:

$$\delta_j = \lambda y_j (1 - y_j) \tag{11}$$

Where:

- δ_j is the success probability increase for student j in the new group
- y_j is the initial success probability prediction for this student by the unmodified model ($0 \leq y_j \leq 1$)

- λ is the multiplying factor for this increase (for the whole set of students in the new group)

This definition makes sure that success probabilities around 0.5 will increase most (note that $y_n(1-y_n)$ maximizes at $y_n = 0.5$) , and probabilities near 0 or 1 will get no increase at all. The rationale behind this definition is the hypothesis that interventions will likely increase success probabilities of students on the verse of success/failing (probabilities around 0.5) the most as exactly this group will benefit from a possible intervention effect. The parameter λ will determine the strength of this effect for the whole group: if lambda is close to zero, there is no significant increase in the success probabilities for all students.

The parameter λ is determined in the sampling process using the historical success ratios. A representation of the model is shown in figure 6.4.

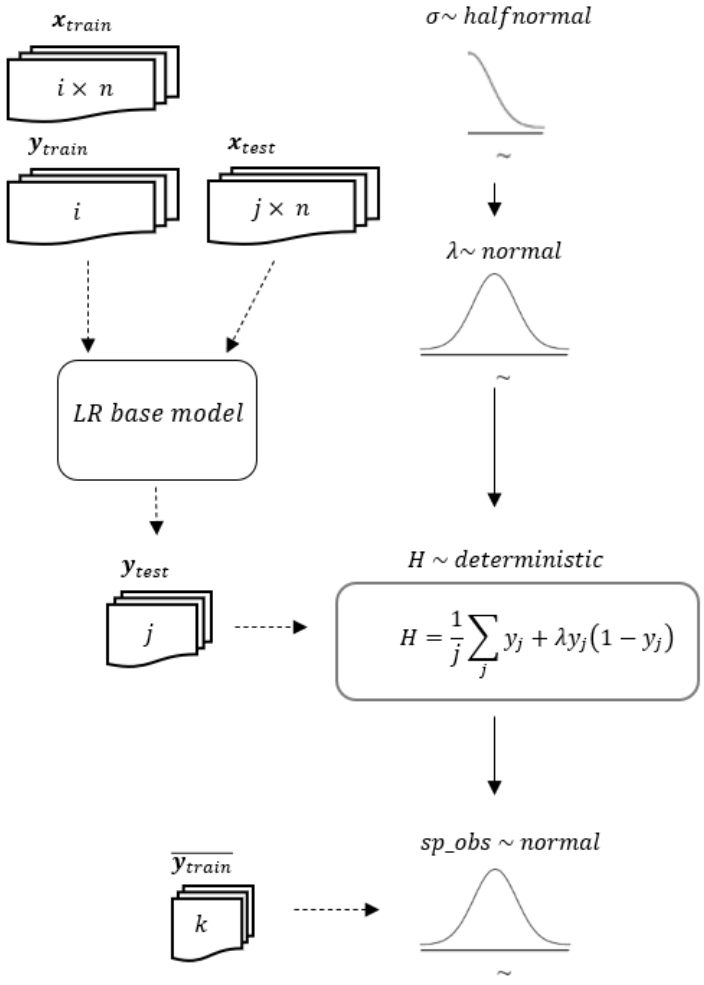

Figure 6.4: Implementation of the Bayesian Lambda Model

In this model, the following parameters are incorporated:

- λ
  The multiplying factor for the variational increase in success probability. Normally distributed with mean 0 and standard deviation of σ.

- σ
  Standard deviation of λ, the prior is set as a halfnormal distribution with standard deviation 1.

- H
  Deterministic variable indicating the success ratio of the new group of students. This variable is calculated as the mean value of the sum of the initial predicted probabilities and the probability increase determined by λ and equation 11.

- sp_{obs}
  Probability distribution of the success ratio. This distribution is modelled as a normal distribution with mean H and the historical group success ratios as observed values. The standard deviation is set equal to that of the historical group success ratios.

For our predictive model, the individual success probability of students in the new group are simply calculated by using the initial probability y_j for student j in the new group, increased with δ_j (equation 11) using the mean of λ determined in the sampling process.

Multiplying factor λ will be close to zero if the initially predicted success ratio is close to the historical group success ratios. The correction is increased most when this initial prediction is significantly different compared to the historical success ratios, and the variance of these ratios is low. We expect the performance of this model to be comparable to the Bayesian Group model previously discussed.

7 Results

In the preceding section, different models were established containing strategies for dealing with the group effect. All models are tested using the synthetic data simulations discussed in section 5, and the Brier score is calculated to rank model performance. Furthermore, the case study data of our example school is applied to the models to examine model performance in a real life scenario.

7.1 Synthetic data simulations

The simulations with increasing GPF are performed for the basic ML models (SQ1), the Group Difference model (SQ3) and the Bayesian models (SQ4). The results are plotted in a diagram indicating the Brier score at different strengths of the group effect. Each datapoint represents the resulting mean Brier scores for 200 simulations. For all models, a polynomial with order 2 is fitted through the points to better illustrate the relation.

7.1.1 Basic ML models (SQ1)

The results of the models created for SQ1, discussed in section 6.1, are shown in figure 7.1.

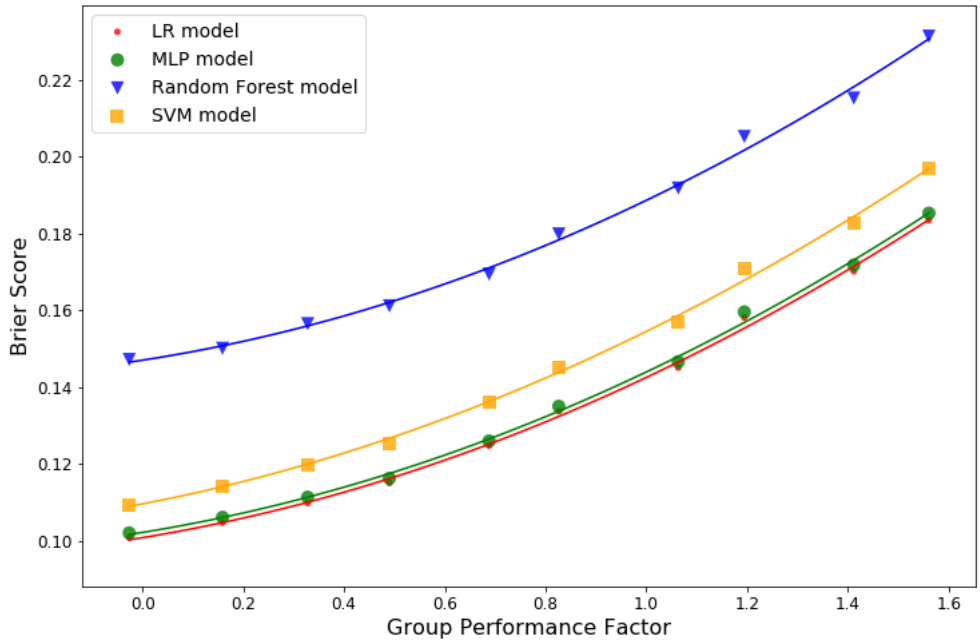

Figure 7.1: Brier scores for the different base models with increasing group effect

The Group Difference model (SQ3) and the Bayesian models (SQ4) will be compared to the best performing model in this set. As the Logistic Regression (LR) model presents the lowest Brier scores for all variations of the GPF, the results of this base model will be plotted in the other diagrams. Obviously, if a model

strategy results in a better performance compared to the LR base model, it will perform better than all the other base models.

We should note that if the results of the same model are plotted again in this section to compare with a new strategy, the simulations for that model were usually repeated. As there is some randomness involved in the generation of data within the simulations, the exact position of datapoints may slightly differ for the same model. However, these differences are minor and the polynomial that is fitted through the datapoints will be similar.

7.1.2 Group Difference model (SQ3)

In this strategy, features relative to the own group are used, as discussed in section 6.3. The results of this model for the synthetic dataset simulations, plotted along the base LR model, are shown in figure 7.2.

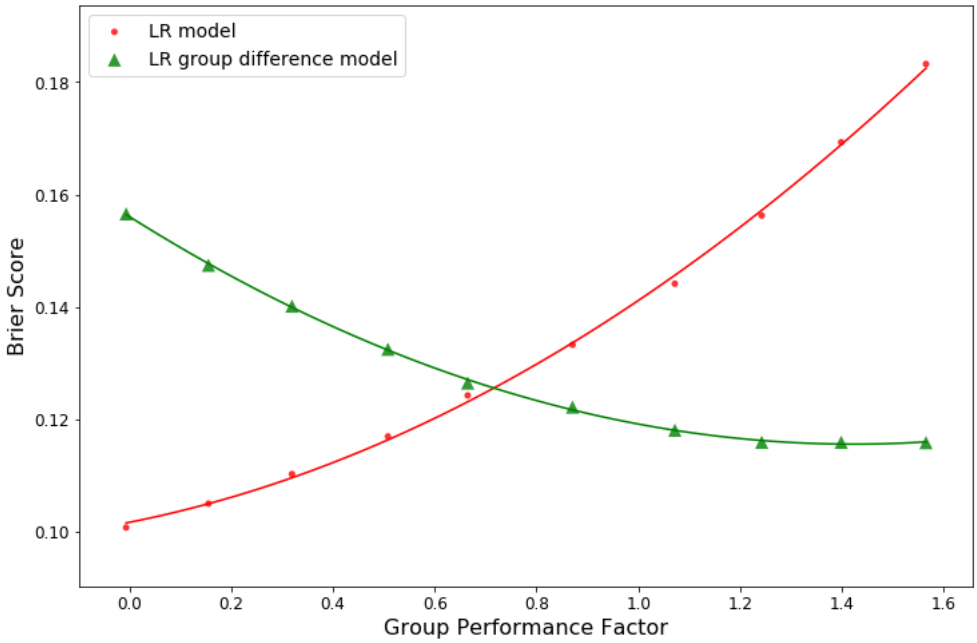

Figure 7.2: Brier scores for the group difference model (relative features) compared to the base LR model

7.1.3 Bayesian models (SQ4)

We created three types of Bayesian models, as discussed in section 6.4. The first one is the Bayesian variant of the Logistic Regression model. This model was included not as a strategy, but to verify the design of our Bayesian modelling techniques. The results of this model, compared to the frequentist LR model, are shown in figure 7.3.

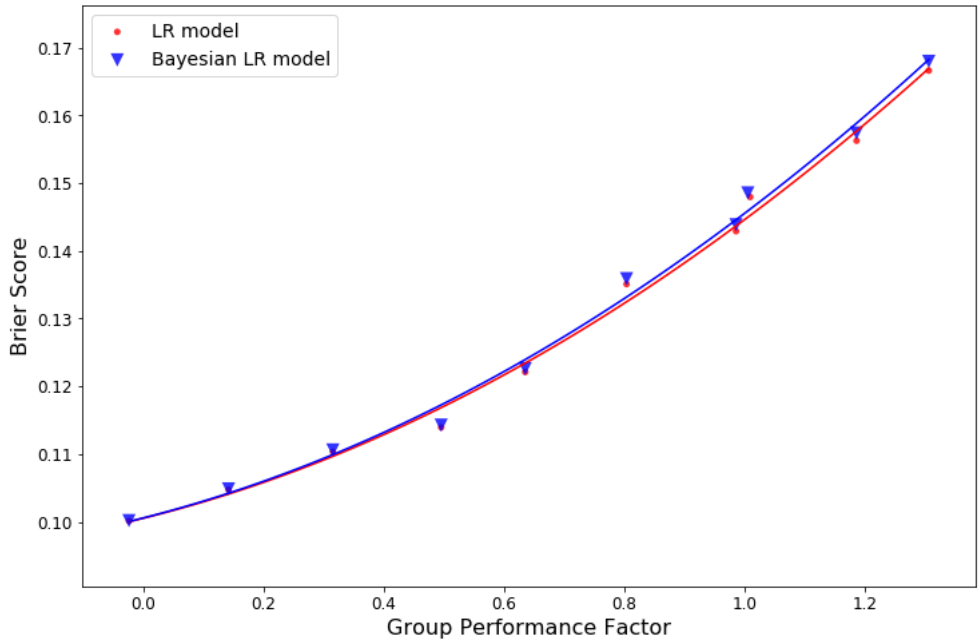

Figure 7.3: Brier scores for the Bayesian Logistic Regression model compared to the frequentist LR model

The first real Bayesian strategy is the Bayesian Group model including a probability distribution for the success ratio of groups. The results of this model, plotted along the base LR model and the strategy of using relative features (group difference model), are shown in figure 7.4.

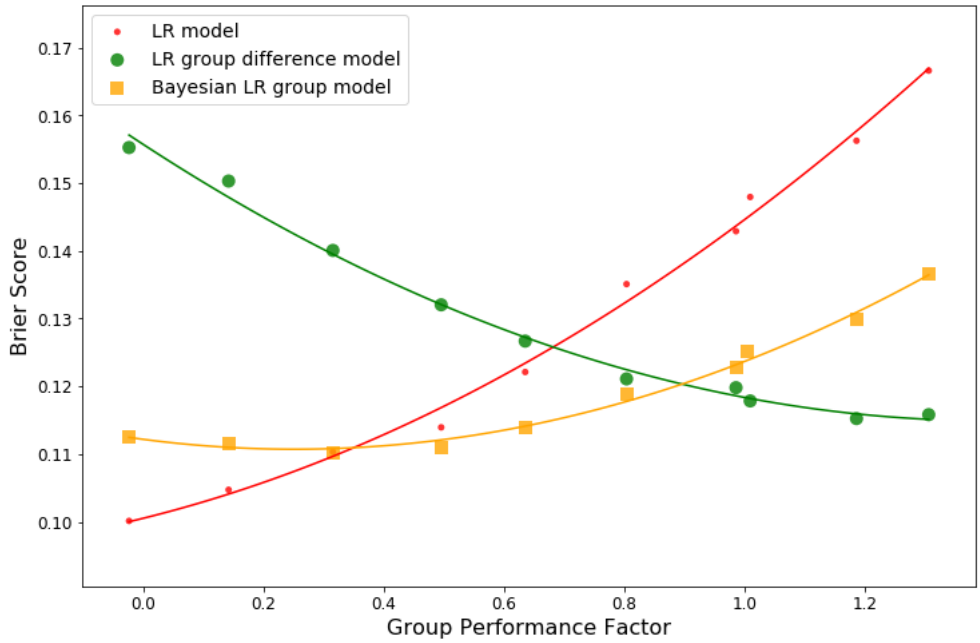

Figure 7.4: Brier scores for the Bayesian Group model compared to the base LR model and the group difference model

The last strategy is the Bayesian Lambda model, creating an improvement in success probability based on success ratios of historical groups. The results of this model, plotted along the base LR model and the Group Difference model, are shown in figure 7.5.

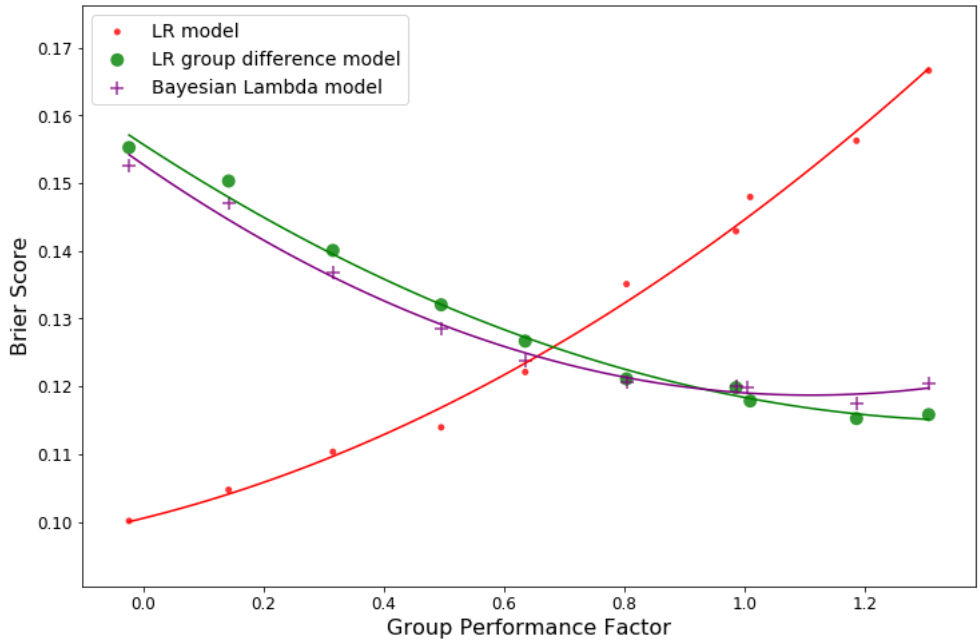

Figure 7.5: Brier scores for the Bayesian Lambda model compared to the base LR model and the group difference model

7.2 Case Study

After running the set of extensive simulations to test the models, they are applied to the case study dataset regarding student success for our example school. As discussed in section 5.2, model performance (Brier score) for the department *H2* will be evaluated, as well as the total Brier score for all departments in transition year 2018. In addition to the models tested using the synthetic data, in this case study the strategy of including more features (SQ2) will also be tested.

7.2.1 Basic ML models (SQ1)

First, the results for the basic Machine Learning models are applied to the case study dataset with the transition year 2018 to be predicted. For reference, two trivial models are included. In the first, all students are promoted. Note that the accuracy of such a model is not too bad, as most of the students are promoted in each department anyway (table 3.2). In the second trivial model, all students will be assigned the *mean success probability* for that department in the training data.

The results for these trivial models and the basic ML models are shown (sorted best-to-worst) in table 7.1 (department *H2*) and table 7.2 (all departments).

Model	Brier score
Logistic Regression	0.077
Support Vector Machine	0.085
Random Forest	0.092
All students mean probability	0.093
Every student promoted	0.104

Table 7.1: Case Study results for the basic ML models (department H2, transition year 2018, best-to-worst)

Model	Brier score
Logistic Regression	0.057
Support Vector Machine	0.064
Random Forest	0.067
All students mean probability	0.091
Every student promoted	0.101

Table 7.2: Case Study results for the basic ML models (all departments, transition year 2018, best-to-worst)

7.2.2 Adding more features (SQ2)

The strategy we did not analyse in the synthetic dataset simulation is including more features in the models. As discussed in section 6.2, some information relatively independent on the course marks is included in the fitting process of the model. The best performing basic models are included, both using only course mark aggregates and the variant with more features. The results of these models are shown (sorted best-to-worst) in table 7.3 (department *H2*) and table 7.4 (all departments).

Model	Brier score
Logistic Regression	0.077
Random Forest (additional features)	0.079
Logistic Regression (additional features)	0.083
Support Vector Machine	0.085
Support Vector Machine (additional features)	0.086
Random Forest	0.092

Table 7.3: Case Study results for the basic models, and the same models with additional features (department H2, transition year 2018, best-to-worst)

7.2.3 Group Difference model (SQ3)

In this strategy, course mark aggregates relative to those of the own group were used. The results for the Logistic Regression and Support Vector Machine models using this group difference, compared to the basic variants of these models, are shown in table 7.5 (department *H2*) and table 7.6 (all departments).

Model	Brier score
Logistic Regression	0.057
Logistic Regression (additional features)	0.057
Random Forest (additional features)	0.060
Support Vector Machine (additional features)	0.062
Support Vector Machine	0.064
Random Forest	0.067

Table 7.4: Case Study results for the basic models, and the same models with additional features (all departments, transition year 2018, best-to-worst)

Model	Brier score
Logistic Regression (Group difference)	0.053
Logistic Regression	0.077
Support Vector Machine (group difference)	0.079
Support Vector Machine	0.085

Table 7.5: Case Study results for the Group Difference models, and the corresponding basic models (department H2, transition year 2018, best-to-worst)

Model	Brier score
Logistic Regression (Group difference)	0.054
Logistic Regression	0.057
Support Vector Machine	0.064
Support Vector Machine (Group Difference)	0.065

Table 7.6: Case Study results for the Group Difference models, and the corresponding basic models (all departments, transition year 2018, best-to-worst)

7.2.4 Bayesian models (SQ4)

Three Bayesian models were tested: the Bayesian variant of the Logistic Regression model and the two strategies (the Bayesian Group model and the Bayesian Lambda model).
The results of these Bayesian models, compared to the frequentist Group Difference model and the base LR model, are shown in table 7.7 (department *H2*) and table 7.8 (all departments).

Model	Brier score
Bayesian Group Model	0.051
Logistic Regression (Group Difference)	0.053
Bayesian Lambda Model	0.055
Logistic Regression	0.077
Bayesian Logistic Regression	0.080

Table 7.7: Case Study results for the Bayesian models, compared to the other models (department H2, transition year 2018, best-to-worst)

Model	Brier score
Logistic Regression (Group difference)	0.054
Bayesian Group Model	0.055
Bayesian Lambda Model	0.056
Logistic Regression	0.057
Bayesian Logistic Regression	0.058

Table 7.8: Case Study results for the Bayesian models, compared to the other models (all departments, transition year 2018, best-to-worst)

All results will be analysed in the next section, including a discussion of the statistical significance.

8 Discussion

In the previous section, the results of our models were presented without comments. In this section, we will discuss both the results of the synthetic dataset simulations, and the model performance with respect to the case study of student success prediction. Furthermore, the significance of the case study results are discussed.

8.1 Synthetic dataset simulations

In the analysis of the basic machine learning models, the (exponential) increase in Brier score with an increasing group effect (GPF) is expected (figure 7.1). The performance of the Random Forest model seems off compared to the other models. However, note that the Brier score is used for comparing the models. The decision tree ensemble provides a probability as the fraction of individual trees predicting a positive class label compared to the total number of decision trees. It is not too surprising that this method produces uncalibrated probabilities and inferior Brier scores. The number of decision trees in the ensemble may be increased (we used 100 trees), but with the number of datapoints and features involved this will probably only result in redundant trees and might not increase the Brier score.

The first strategy dealing with the group effect is the Group Difference model (figure 7.2). This model clearly outperforms the base LR model if the GPF is significant, but the performance is significantly worse in the absence of a group effect, or when this effect is small. The turning point is a GPF of about 0.7. Note that this is substantial: if the *group effect* is 2 (the naive prediction of the new group is two standard deviations lower compared to the training groups) the *intervention effect* is 0.35. This essentially means that more than a third of the students in the new group would have to be promoted despite a negative prediction of the base model. In the case of smaller group effects, the LR base model performs better than the Group Difference model.

The first Bayesian model presented is the Bayesian Logistic Regression model (figure 7.3). As expected, the performance is similar to that of the frequentist LR model. Since we used *weak priors* for the β parameters, these values are fitted only on the training dataset, which is exactly the case for the frequentist model. The similar relation, however, provides confidence in the Bayesian modelling techniques and the chosen sampling algorithm, tuning parameters and sample size. This is useful as the more complex Bayesian models don't have frequentist counterparts.

The strategy used in the Bayesian Group Model (figure 7.4) turns out to be a robust one: the Brier score is quite stable for all variations of the GPF. One might expect this model to perform in between the unmodified (LR) model and the Group Difference model, but the Brier score constantly approaches the best of the two for all group effect variations. In a substantial part ($0.35 < GPF < 0.9$) the Bayesian Group model even outperforms both other models. Only in the case of a small group effect or a very strong GPF, the model is outperformed by the base LR model and the Group Difference model, respectively. We should

keep in mind however that in practice, the *intervention effect* is not known beforehand, and should be estimated by domain experts. In these uncertain cases, it is less risky to use a robust model, and the Bayesian Group model might be a proper candidate in these scenarios.

The second Bayesian strategy, used in the Bayesian Lambda model, seems comparable to the Group Difference model. This is somewhat unexpected, as we assumed this model to perform more like the Bayesian Group model (noted in section 6.4). In retrospect, however, this behaviour is not too surprising. In the Group Difference model, the features are expressed as difference from the group mean (relative features). Let us now simplify the scenario to one feature: the mean course mark. If the new group performs less than the training groups, the transformation essentially means a shift of the 'S-curve' fitted in a Logistic Regression model as shown in figure 8.1. This shift results in an increased success probability for all students. However, as the slope of the logistic function is highest in the middle, students on the edge of failing/passing (an initial success probability around 0.5) "profit" more of this shift, e.g. their success probability is increased most. This is exactly the assumption we used for the Bayesian Lambda model. Apparently, both models use a similar mechanism, which is confirmed by their comparable Brier scores in our synthetic dataset simulations.

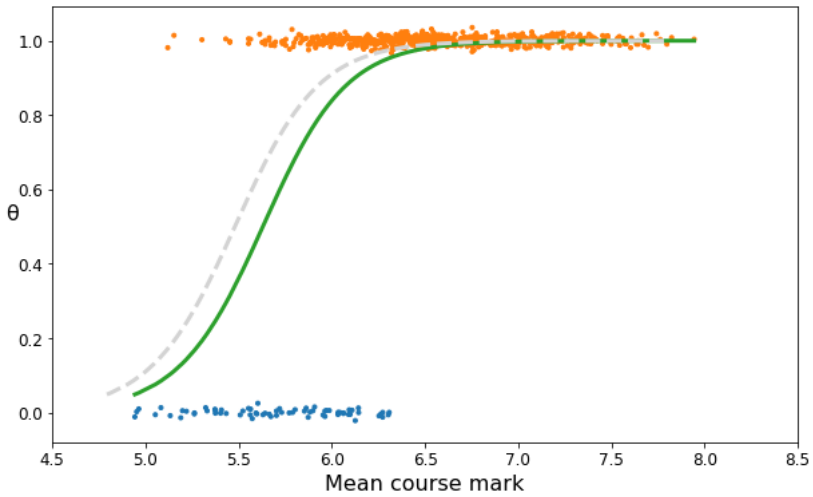

Figure 8.1: Illustration of the shift of the S-curve in a Logistic Regression model. The shift of the green line to the dashed grey line increases θ (success probability) for all students, with the largest increase in the middle (for the slope of the logistic function maximizes there)

8.2 Case Study

For the most part, the case study results correspond to the results of the synthetic data simulations. In the results of the basic models (table 7.1 and 7.2), the Machine Learning models fortunately perform better than the trivial models (the models where all students get the mean training probability or every

student is promoted). The trivial models may appear to do quite well. However, note that because of the imbalance in the data (most students get promoted), predicting all students as successful results in a better than random prediction.

The considerably higher Brier scores in the *H2* department compared to the scores for all departments may indicate the strong group effect at this department in the transition year 2018. All Machine Learning models suffer from this effect. As expected, this is not the case for the trivial models. Since the *intervention effect* makes sure that the success ratio of the new group approaches the historical success ratios despite degrading course marks, the trivial models perform comparable in both situations.

When adding more features (table 7.3 and 7.4), only the Random Forest model evidently improves. This is probably due to the mechanism of the decision tree ensemble. In the process of creating individual decision trees, both features and datapoints are split: different decision trees use different features. When adding more features, this model may improve as there are now less redundant trees and the ensemble will be stronger. The model approaches but does not outperform the basic Logistic Regression model.

In general, it appears that adding these features does not improve the models (the performance may even decline). Apparently, the predictive value of the added features is minor compared to the features based on student course marks, which may be explained my domain knowledge. In the student course marks in the beginning of the schoolyear, all underlying causes are already incorporated. Course marks result from intelligence, dedication, social-emotional circumstances, class absence, teacher quality and many other factors. Including quantified features with respect to these underlying causes does not add predictive value, apparently.

The Group Difference model, using continuous features relative to the group mean (SQ3), outperformed the base models only in the case of substantial group effects, e.g. a high GPF. This behaviour appears to be confirmed by the results of the case study. The difference between the relative model (Group difference) and the unmodified model is clear when department *H2* is analysed (table 7.7). The effect is still visible when all departments are concerned (table 7.8). However, this somewhat minor difference could solely be due to the problematic department *H2*, as that is part of the whole set of departments too. In this case, the performance of the model is less for other departments, where there is no or only a small group effect.

These results appear to be in line with the simulations. However, this is not completely true. Note that the GPF calculated for department *H2* in the case study was 0.38 (table 5.1). While this is a considerable group effect, we do not expect the Group Difference model to outperform the base model at this GPF based on the simulations (figure 7.2). Here, we noticed the turning point at a GPF of about 0.7, which is almost two times higher than the group effect concerned. There may be two reasons for this contradiction. The Group Performance Factor we defined in equation 9, may not be universal. In this case, it would be incorrect to compare group factors in different situations. Another

reason may simply be the fact that in the case study only one "simulation" is concerned, which may contain too much variance to confirm the behaviour as shown in the synthetic dataset results.

Regarding the results of the Bayesian models (table 7.7 and 7.8) we may first note that the Bayesian variant of the Logistic Regression model is indeed similar to that of the frequentist model, the Brier scores are not that different corresponding to the simulation results.
The first Bayesian strategy incorporated in the Bayesian Group model shows the best Brier score in the case study with respect to department *H2*. This corresponds to our findings in the synthetic dataset simulations, where the Bayesian Group model outperformed the other models in small to medium group effects (including the GPF of 0.38). However, it is not evident that the difference in Brier scores between the three strategies are statistically significant.
This observation holds even more for the analysis of the models with respect to all departments (table 7.8): the Brier scores seem very close. Though these are mean scores for a larger group of individuals and a smaller difference is expected, we should be careful to draw conclusions based on these results.

8.3 Statistical Significance

To provide further insight in the reliability of the case study results, the confidence intervals of the Brier scores presented in tables 7.7 and 7.8 are determined. Since we cannot assume that Brier scores for individual students within a model are normally distributed, the *Bootstrap Method* was used to establish these intervals. In this method, sampling with replacement is used to estimate a parameter from a distribution [Efron, 1981], in this case the Brier score of the different models.

The results of the Bootstrap sampling (1E5 samples for each model) to determine the confidence intervals for the Brier scores of the different models in the case study are shown in tables 8.1 (department *H2* only) and 8.2 (all departments). The 2.5% and 97.5% percentiles constitute the range of the 95% confidence interval.

Model	**Mean**	**2.5% perc.**	**97.5% perc.**
Bayesian Group Model	0.051	0.034	0.070
Group Difference Model	0.053	0.033	0.076
Bayesian Lambda Model	0.055	0.032	0.081
Logistic Regression	0.077	0.052	0.104
Bayesian Logistic Regression	0.080	0.054	0.109

Table 8.1: 95% confidence intervals for the Brier score of the different models resulting from Bootstrap sampling (department H2, transition year 2018)

From these intervals, the uncertainty of model performance in this case study is clear. After all, this is the reason we used the mean of 200 simulations to determine the Brier score at different group effects in the synthetic dataset simulations.

Model	Mean	2.5% perc.	97.5% perc.
Group difference model	0.054	0.046	0.063
Bayesian Group Model	0.055	0.047	0.063
Bayesian Lambda Model	0.056	0.047	0.064
Logistic Regression	0.057	0.049	0.065
Bayesian Logistic Regression	0.058	0.049	0.066

Table 8.2: 95% confidence intervals for the Brier score of the different models resulting from Bootstrap sampling (all departments, transition year 2018)

For this case study, though results appear to be in line with the synthetic dataset results, we are unable to determine a significant 'winning' model.

9 Conclusion

Using the results and discussion of the previous sections, we may draw some conclusions with respect to the research topic. Furthermore, the recommendations for the use of models are provided, and possible future work is discussed.

9.1 Conclusions and recommendations

Considering the research question, the corresponding sub questions and the results of our simulations and the case study, the following conclusions may be derived with respect to the domain of using Machine Learning for student success prediction at secondary schools in the Netherlands:

1. Using unmodified Machine Learning models based on early student course marks may fail to provide a reliable prediction because of the Group Performance Effect.
2. In this particular case, using additional features in these models does not improve results, which may be explained by the fact that course marks already result from a wide range of causes.
3. If there is a very strong group effect, e.g. a high Group Performance Factor, using features relative to the own group (the Group Difference Model) is a successful strategy in dealing with this effect.
4. Two Bayesian strategies appear to improve results. The robust Bayesian Group model improves results at moderate group effect strengths, while the Bayesian Lambda model may be used with strong group effects, comparable to the Group Difference Model.

Not all strategies are effective in all scenarios, and a thorough consideration should be made for the selection of a certain model. We recommend the following procedure for creating predictive models in this domain:

- When no significant Group Performance Effect is expected, use a standard ML model like Logistic Regression.
- If a very strong group effect is expected, use the frequentist Group Difference model or the Bayesian Lambda model
- In all other cases, including the scenario when estimating the group effect by domain experts is not available, use the robust Bayesian Group Model

In conclusion, we were able to successfully improve student success prediction by dealing with the Group Performance Problem using Machine Learning, with different strategies. These strategies may be used *before* the actual class labels of the new group of students are known, which is a useful contribution to the field of dataset shift and adapting algorithms. Using extensive simulations, the effect of the different strategies on model performance was clear. The case study using data from our example school appears to support these conclusions, but the results were not significant.

9.2 Future work

Future work may focus on the definition of the GPF, the variable quantifying the strength of a group effect in a certain scenario. From our results, it is not evident that this factor is universal. It may we worth investigating if there is a non-dimensional constant describing the strength of a group effect in all scenarios. Using this factor, the strategies developed and future strategies may be thoroughly tested and compared to each other. As the effect consists of two parts: the *group effect* and the *intervention effect*, it may be necessary to separate these effects in multiple constants. Furthermore, note that in our simulations only the intervention effect was varied. It may be helpful to examine the effect of varying both parts of the GPF separately.

The strategies and corresponding models developed in this domain should be tested in other scenarios. In all problems dealing with a group effect, e.g. if the GPE definition of section 3 holds, these strategies may improve the results of predictive models. More strategies for dealing with the group effect may also be formulated and examined. For example, one Bayesian variant we did not analyse was using the uncertainty of the individual prediction by a Bayesian LR model to determine the individual increase in success probability, instead of the quadratic function used in the Bayesian Lambda model.

Finally, in some scenarios the runtime of predictions may be important. In our case of student success prediction, this was no issue, but it may be useful to evaluate different models with respect to the time needed to provide a proper prediction. After all, the sampling process used in our Bayesian models was converging fast and was able to provide mean parameter values with an acceptable accuracy in less than a minute, but this may be too long for certain scenarios, i.e. online learning. Furthermore, in problems with a higher number of covariates (features) or larger datasets, the running time of predictive models may be more relevant. Future work with respect to this attribute is recommended.

10 References

Scientific References

[Bach and Maloof, 2008] Bach, S. and Maloof, M. (2008). Paired learners for concept drift. pages 23–32.

[Bifet et al., 2010] Bifet, A., Holmes, G., and Pfahringer, B. (2010). Leveraging bagging for evolving data streams. In Balcázar, J. L., Bonchi, F., Gionis, A., and Sebag, M., editors, *Machine Learning and Knowledge Discovery in Databases*, pages 135–150, Berlin, Heidelberg. Springer Berlin Heidelberg.

[Brier, 1950] Brier, G. W. (1950). Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78(1):1–3.

[Daskalakis et al., 2015] Daskalakis, C., Diakonikolas, I., and Servedio, R. A. (2015). Learning poisson binomial distributions. *Algorithmica*, 72(1):316–357.

[Efron, 1981] Efron, B. (1981). Nonparametric estimates of standard error: The jackknife, the bootstrap and other methods. *Biometrika*, 68(3):589–599.

[Gama et al., 2014] Gama, J. a., Žliobaitundefined, I., Bifet, A., Pechenizkiy, M., and Bouchachia, A. (2014). A survey on concept drift adaptation. *ACM Comput. Surv.*, 46(4).

[Homan and Gelman, 2014] Homan, M. D. and Gelman, A. (2014). The no-u-turn sampler: Adaptively setting path lengths in hamiltonian monte carlo. *J. Mach. Learn. Res.*, 15(1):1593–1623.

[Hulten et al., 2001] Hulten, G., Spencer, L., and Domingos, P. (2001). Mining time-changing data streams. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '01, page 97–106, New York, NY, USA. Association for Computing Machinery.

[Lu et al., 2019] Lu, J., Liu, A., Dong, F., Gu, F., Gama, J., and Zhang, G. (2019). Learning under concept drift: A review. *IEEE Transactions on Knowledge and Data Engineering*, 31(12):2346–2363.

[Merkle and Steyvers, 2013] Merkle, E. and Steyvers, M. (2013). Choosing a strictly proper scoring rule. *DECISION ANALYSIS*, 10(4):292–304.

[Moreno-Torres et al., 2012] Moreno-Torres, J. G., Raeder, T., Alaiz-Rodríguez, R., Chawla, N. V., and Herrera, F. (2012). A unifying view on dataset shift in classification. *Pattern Recognition*, 45(1):521–530.

[Niculescu-Mizil and Caruana, 2005] Niculescu-Mizil, A. and Caruana, R. (2005). Predicting good probabilities with supervised learning. In *Proceedings of the 22nd International Conference on Machine Learning*, ICML '05, page 625–632, New York, NY, USA. Association for Computing Machinery.

[Pedregosa et al., 2011] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and

Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

[Platt, 1999] Platt, J. C. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *ADVANCES IN LARGE MARGIN CLASSIFIERS*, pages 61–74. MIT Press.

[Salvatier et al., 2016] Salvatier, J., Wiecki, T. V., and Fonnesbeck, C. (2016). Probabilistic programming in python using pymc3. *PeerJ Computer Science*, 2:e55.

[Xue and Titterington, 2008] Xue, J.-H. and Titterington, D. M. (2008). Comment on "on discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes". *Neural Processing Letters*, 28(3):169.

Other References

[Claas L., 2017] Claas L., Janssen W., N. W. (2017). Marktscandata-analyse tooling. Market scan report, Kennisnet, funded by the Dutch Ministry of Education, Culture and Science. Available at https://www.kennisnet.nl/fileadmin/kennisnet/onderwijsvernieuwing/Documenten/Kennisnet-Markscan-data-analyse-tooling.pdf.

[Korb, 2011] Korb, K. B. (2011). Chapter 1 - bayesian reasoning and chapter 2 - introducing bayesian networks. In *Bayesian Artificial Intelligence*, pages 12, 29. CRC Press, Oxford.

[Lee and Wagenmakers, 2014] Lee, M. D. and Wagenmakers, E.-J. (2014). *Bayesian Cognitive Modeling: A Practical Course*. Cambridge University Press.

[Onderwijsinspectie, 2018] Onderwijsinspectie (2018). Onderwijsresultaten Voortgezet Onderwijs 2018. Technical explanation, Inspectorate of Education, Dutch Ministry of Education, Culture and Science. Available at https://www.onderwijsinspectie.nl/documenten/rapporten/2018/05/15/onderwijsresultaten-2018-technische-toelichting.

[Theodoridis, 2015] Theodoridis, S. (2015). Chapter 1 - introduction. In Theodoridis, S., editor, *Machine Learning*, pages 1 – 8. Academic Press, Oxford.

11 List of abbreviations

i.i.d.	independent and identically distributed
GPF	Group Performance Factor
GPP	Group Performance Problem
LR	Logistic Regression
MCMC	Monte Carlo Markov Chain
ML	Machine Learning
NUTS	No-U-Turn Sampler
PE	Primary Education
RF	Random Forest
RQ	Research Question
SAS	School Administration System
SVM	Support Vector Machine
SQ	Sub Question